

សាកលវិទ្យាល័យភូមិន្ទនីតិសាស្ត្រ
និង វិទ្យាសាស្ត្រសេដ្ឋកិច្ច

សារណាបញ្ចប់ការសិក្សា

ប្រព័ន្ធគ្រប់គ្រងសណ្ឋាគារ

ស្រាវជ្រាវចាប់ពីថ្ងៃទី១៦ ខែមីនា ឆ្នាំ២០២០ ដល់ថ្ងៃទី១៥ ខែឧសភា ឆ្នាំ២០២០

តាក់តែងឡើងដោយ

និស្សិតឈ្មោះ

ឡាំ សុ ផ

គុន សកល

សាស្ត្រាចារ្យណែនាំ

បណ្ឌិត **ទិន ហេង**

ថ្នាក់បរិញ្ញាបត្រ សេដ្ឋកិច្ចព័ត៌មានវិទ្យា

ជំនាន់ទី ១៥

ឆ្នាំចូលសិក្សា

២០១៦

ឆ្នាំសរសេរសារណា ២០២០

សេចក្តីថ្លែងអំណរគុណ

យើងខ្ញុំ ឡាំ សុផ និងគុន សកល ជានិស្សិតជំនាន់ទី១៥ ផ្នែកសេដ្ឋកិច្ចព័ត៌មានវិទ្យា នៃ សាកលវិទ្យាល័យភូមិន្ទនីតិសាស្ត្រ និងវិទ្យាសាស្ត្រសេដ្ឋកិច្ចឆ្នាំ ២០២០ -២០២១។ យើងខ្ញុំទាំងពីរ សូមធ្វើការគោរព និងថ្លែងអំណរគុណប្រកបដោយកត្តញ្ញតាមម៉ាយ៉ាងជ្រាលជ្រៅជូនចំពោះ ៖

សូមថ្លែងអំណរគុណយ៉ាងជ្រាលជ្រៅចំពោះលោកឪពុក និងអ្នកម្តាយ ដែលលោកទាំងពីរ បានផ្តល់កំណើត និងខិតខំចិញ្ចឹមបីបាច់ថែរក្សាផ្តល់ទំនុកបំរុងដល់កូនតាំងពីតូចរហូតដល់សព្វថ្ងៃ នេះ មិនត្រឹមតែប៉ុណ្ណោះលោកទាំងពីរបានខិតខំប្រឹងប្រែង ព្យាយាមពន្យល់ អប់រំទូន្មានកូនឲ្យក្លាយ ជា កូនល្អ សិស្សល្អ និងពលរដ្ឋល្អក្នុងសង្គម ។ លើសពីនេះសូមគោរពចំពោះការស្រឡាញ់យ៉ាង ជ្រាលជ្រៅរបស់លោកទាំងពីរចំពោះកូនៗ ចង់ឃើញកូនមានអនាគតល្អ និង ចំណេះដឹងជ្រៅជ្រះ។

សូមថ្លែងអំណរគុណយ៉ាងជ្រាលជ្រៅចំពោះក្រសួងអប់រំ យុវជន និងកីឡា ដែលបានផ្តល់ អាហាររូបករណ៍ដល់រូបយើងខ្ញុំតាំងពីឆ្នាំទី១ រហូតដល់បញ្ចប់ឆ្នាំទី ៤។

សូមថ្លែងអំណរគុណយ៉ាងជ្រាលជ្រៅចំពោះលោកឯកឧត្តម សាកលវិទ្យាធិការ សាកល វិទ្យាធិការរង និងសាស្ត្រាចារ្យទាំងអស់ដែលបានអនុញ្ញាតឱ្យយើងខ្ញុំបានសិក្សារហូតបានបញ្ចប់ ថ្នាក់បរិញ្ញាបត្រសេដ្ឋកិច្ចព័ត៌មានវិទ្យានេះ។

សូមថ្លែងអំណរគុណយ៉ាងជ្រាលជ្រៅចំពោះសាស្ត្រាចារ្យទាំងអស់ដែលបានខិតខំប្រឹង ប្រែងផ្តល់ចំណេះដឹង ដល់រូបយើងខ្ញុំដោយយកចិត្តទុកដាក់ ជាពិសេសគឺជួយអប់រំទូន្មានដើម្បីក្លាយ ទៅជាពលរដ្ឋល្អក្នុងសង្គម។

សូមថ្លែងអំណរគុណយ៉ាងជ្រាលជ្រៅចំពោះ លោកសាស្ត្រាចារ្យបណ្ឌិត **ទិន ហេង** ជាសា ស្ត្រាចារ្យក្នុងការដឹកនាំសរសេរសារណានេះឡើង។ លោកគ្រូបានចំណាយពេលវេលាដ៏មានតម្លៃ ក្នុងការបង្ហាត់បង្រៀន ផ្តល់ជាគំនិត ផ្តល់យោបល់ល្អៗព្រមទាំងឯកសារផងដែរ។ លើសពីនេះទៅ ទៀតលោកបានជួយកែសម្រួលនៅចន្លោះខ្វះខាតពាក្យពេជ័ន៍ និងបែបបទក្នុងការសរសេរសារ ណានេះឡើង។ កត្តាទាំងអស់នេះហើយទើបធ្វើឲ្យការសរសេរសារណាបញ្ចប់ការសិក្សានេះបញ្ចប់ ដោយជោគជ័យ។

ជាចុងក្រោយយើងខ្ញុំសូមលើកម្រាមដៃដប់គោរពថ្លែងអំណរគុណសាជាថ្មីម្តងទៀត ជូន ចំពោះអ្នកមានគុណទាំងពីរ ឯកឧត្តម លោក លោកស្រី សាស្ត្រាចារ្យ ដូចដែលបានរៀបរាប់ខាងលើជូ បតែសំណាងល្អ ទទួលនូវសេចក្តីសុខគ្រប់ប្រការ និង ជោគជ័យគ្រប់ភារកិច្ចការងារ ។

អារម្ភកថា

ក្នុងនាមយើងខ្ញុំជានិស្សិត ដែលត្រូវបញ្ចប់ការសិក្សាថ្នាក់បរិញ្ញាបត្រផ្នែកសេដ្ឋកិច្ចព័ត៌មាន វិទ្យាជំនាន់ទី ១៥ នៃសាកលវិទ្យាល័យភូមិន្ទនីតិសាស្ត្រ និងវិទ្យាសាស្ត្រសេដ្ឋកិច្ចមានសេចក្តីរីករាយ យ៉ាងខ្លាំងដែលសាកលវិទ្យាល័យបានអនុញ្ញាតឲ្យយើងខ្ញុំសរសេរសារណាបញ្ចប់ការសិក្សានេះឡើង។ មិនត្រឹមតែប៉ុណ្ណោះ ការចងក្រងសារណានេះបានធ្វើឲ្យការសិក្សា និងសមត្ថភាពយើងខ្ញុំកាន់តែ រីកចម្រើនយ៉ាងខ្លាំង ជាពិសេសគឺទទួលបាននូវបទពិសោធន៍បន្ថែមទៀតលើការអនុវត្តជាក់ស្តែង។

សៀវភៅសារណាបញ្ចប់ការសិក្សាប្រធានបទ៖ **ការបង្កើតប្រព័ន្ធគ្រប់គ្រងសណ្ឋាគារ** នេះត្រូវបានចងក្រងឡើងដោយប្រមូលផ្តុំនូវ ទ្រឹស្តីនៃបច្ចេកវិទ្យាមួយចំនួនដែលប្រើប្រាស់ដើម្បី បង្កើតកម្មវិធី និងបង្ហាញពី ដំណើរការនៃការបង្កើតកម្មវិធីតាមលំដាប់លំដោយ។ ជាពិសេសមាន ពន្យល់ពីរបៀបប្រើប្រាស់មួយចំនួនក្នុងកម្មវិធី ដើម្បីផ្តល់ភាពងាយស្រួលអាន និងងាយយល់។

ខ្ញុំសង្ឃឹមថាសៀវភៅនេះនឹងអាចផ្តល់ជាជំនួយស្មារតី ក៏ដូចជាមេរៀន ឬគំរូដ៏មានសារ ប្រយោជន៍ដល់ម្ចាស់និស្សិតជំនាន់ក្រោយ។ ទុកសម្រាប់សិក្សាស្វែងយល់ និងពង្រឹងស្មារតីក្នុងការ ស្រាវជ្រាវ។ ម្យ៉ាងវិញទៀតយើងខ្ញុំសូមសំណូមពរដល់ម្ចាស់ជំនាន់ក្រោយ។ ឲ្យខិតខំប្រឹងប្រែងរៀន សូត្រ ដើម្បីចូលរួមអភិវឌ្ឍន៍សង្គមជាតិយើងឲ្យមានការរីកចម្រើន និងជឿនលឿនគ្រប់វិស័យ ទាំង វិស័យបច្ចេកវិទ្យា និងវិស័យផ្សេងៗទៀតក្នុងប្រទេសកម្ពុជា។

ក្នុងការចងក្រងសៀវភៅនេះឡើងប្រាកដជាមានចំណុចខ្វះខាត និងកំហុសជាក់ជាពុំខាន ហេតុនេះហើយក្នុងនាមយើងខ្ញុំជាអ្នកចងក្រងសៀវភៅនេះឡើង ក៏សូមអភ័យទោសទុកជាមុនពី សំណាក់មិត្តនិស្សិត និងប្រិយមិត្តអ្នកអានទាំងឡាយចំពោះរាល់កំហុសឆ្គងទាំងឡាយណាដែល កើតឡើងដោយអចេតនា។ យើងខ្ញុំរីករាយស្វាគមន៍ រាល់ការរិះគន់ និងកែលម្អពីសំណាក់មិត្ត និស្សិត និងមិត្តអ្នកអានទាំងឡាយ ក៏ដូចជាលោកគ្រូ អ្នកគ្រូ ចំពោះស្នាដៃនេះ ក្នុងន័យស្ថាបនាដើម្បី ឲ្យស្នាដៃនេះកាន់តែមានភាពល្អប្រសើរ។ ម្យ៉ាងវិញទៀតសូមថ្លែងអំណរគុណយ៉ាងជ្រាលជ្រៅដល់ ប្រិយអ្នកអានទាំងអស់ដែលចំណាយពេលវេលា និងយកចិត្តទុកដាក់អានសៀវភៅនេះ សូមជូនពរ ឲ្យមានសេចក្តីសុខ និងសំណាងល្អគ្រប់ៗគ្នា។

មាតិកា

បញ្ជីអក្សរកាត់.....	v
បញ្ជីរូបភាព.....	vi

សេចក្តីផ្តើម

១ លំនាំបញ្ហានៃការស្រាវជ្រាវ.....	១
២ មូលដ្ឋាននៃការស្រាវជ្រាវ.....	២
៣ ចំណោទបញ្ហានៃការស្រាវជ្រាវ.....	២
៤ គោលបំណងនៃការស្រាវជ្រាវ.....	២
៥ វិសាលភាព និងដែនកំណត់នៃការស្រាវជ្រាវ.....	៣
៦ អត្ថប្រយោជន៍នៃការសិក្សាស្រាវជ្រាវ.....	៣

ជំពូកទី ១

ទ្រឹស្តីពាក់ព័ន្ធនឹងការបង្កើតប្រព័ន្ធ

១.១ ទ្រឹស្តីពាក់ព័ន្ធនឹង Software Engineering	៥
១.១.១ និយមន័យ Software Engineering.....	៥
១.១.២ ម៉ូដែល(Model)	៥
១.១.៣ អ្វីទៅជា Waterfall Model	៦
១.១.៣.១ Requirement Gathering Analysis	៧
១.១.៣.២ System	៧
១.១.៣.៣ Implementation.....	៧
១.១.៣.៤ Testing	៧
១.១.៣.៥ Development of System.....	៨
១.១.៣.៦ Maintenance	៨
១.២ ទ្រឹស្តីពាក់ព័ន្ធនឹងអ៊ុនធឺណិត	៩
១.២.១ ប្រវត្តិសង្ខេបនៃអ៊ុនធឺណិត.....	៩
១.២.២ អ្វីទៅជាអ៊ុនធឺណិត	៩
១.២.៣ អ្វីទៅជា ISP.....	១១
១.៣ ទ្រឹស្តីពាក់ព័ន្ធនឹងគេហទំព័រ	១១
១.៣.១ អ្វីទៅជាគេហទំព័រ	១១

១.៣.២ អ្វីទៅជា Web Page	១១
១.៣.៣ ប្រភេទនៃគេហទំព័រ	១១
១.៣.៣.១ Static Website.....	១១
១.៣.៣.២ Dynamic Website.....	១២
១.៣.៤ លក្ខណៈនៃគេហទំព័រ	១២
១.៣.៤.១ Personal Website	១៣
១.៣.៤.២ Information Website.....	១៤
១.៣.៤.៣ Career Website.....	១៤
១.៣.៤.៤ E-Commerce Website.....	១៥
១.៣.៤.៥ Entertainment & News Website	១៦
១.៣.៥ អ្វីទៅជា Home Page.....	១៦
១.៣.៦ អ្វីទៅជា Domain Name	១៦
១.៣.៧ អ្វីទៅជា Web Browser	១៧
១.៣.៨ អ្វីទៅជា URL.....	១៧
១.៣.៩ អ្វីទៅជា FTP	១៧
១.៤ ទ្រឹស្តីពាក់ព័ន្ធនឹងការកសាងគេហទំព័រ.....	១៨
១.៤.១ អំពី HTML.....	១៨
១.៤.២ អំពី CSS.....	១៩
១.៤.៣ អំពី Bootstrap.....	២០
១.៤.៣.១ របៀបទាញយក Bootstrap មកប្រើប្រាស់.....	២០
១.៤.៣.២ សារប្រយោជន៍នៃ Bootstrap.....	២២
១.៤.៤ អំពី Java Script.....	២២
១.៤.៥ អំពី JQuery.....	២៣
១.៤.៦ អំពី Laravel Framework	២៦
១.៤.៦.១ Server Requirement សម្រាប់ Laravel 6.0	២៦
១.៤.៦.២ Model-View-Controller(MVC).....	២៧
១.៤.៦.៣ របៀប Laravel ធ្វើការ.....	២៧
១.៤.៦.៤ Routing.....	២៨

១.៤.៦.៥ Middleware.....	២៨
១.៤.៦.៦ HTTP Request	៣០
១.៤.៦.៧ HTTP Response	៣១
១.៤.៦.៨ Validator.....	៣១
១.៤.៦.៩ View	៣២
១.៤.៦.១០ Database Configuration	៣២
១.៤.៦.១១ Database Migration	៣៣
១.៤.៦.១២ Database QueryBuilder	៣៤
១.៤.៦.១៣ Eloquent ORM	៣៥
១.៤.៧ មេរៀនសង្ខេប PHP.....	៣៧
១.៤.៧.១ និយមន័យ	៣៧
១.៤.៧.២ អថេរ(Variable).....	៣៧
១.៤.៧.៣ Data Type	៣៧
១.៤.៧.៤ Operator	៣៨
១.៤.៧.៥ Control Statement.....	៤១
១.៤.៧.៦ Function.....	៤៥
១.៤.៧.៧ Array.....	៤៥
១.៥ ទ្រឹស្តីពាក់ព័ន្ធនឹង Database.....	៤៦
១.៥.១ DBMS.....	៤៧
១.៥.១.១ MySQL.....	៤៧
១.៥.១.២ ណែនាំអំពី PHP MyAdmin.....	៤៨
១.៥.២ Data Analysis	៤៨
១.៥.៣ Database Relationship.....	៤៩

ជំពូកទី ២

ដំណើរការនៃការស្រាវជ្រាវ

២.១ វិធីសាស្ត្រស្រាវជ្រាវ	៥០
២.១.១ ការពិភាក្សាដើម្បីជ្រើសរើសប្រធានបទ	៥០

២.១.២ ការជ្រើសរើសសាស្ត្រាចារ្យណែនាំគម្រោង.....	៥០
២.១.៣ ការប្រមូលទិន្នន័យ.....	៥១
២.១.៤ ការកំណត់តម្រូវការព័ត៌មាន.....	៥០
២.១.៥ ការរៀបចំរចនាសម្ព័ន្ធនៃប្រព័ន្ធ.....	៥១
២.១.៦ ការសរសេរកូដ.....	៥១
២.១.៧ ការពិសោធន៍សាកល្បង និងថែរក្សាប្រព័ន្ធ.....	៥២
២.២ គម្រោងពេលវេលានៃការស្រាវជ្រាវ.....	៥២

ជំពូកទី ៣

ការអនុវត្តលើគេហទំព័រ

៣.១ ការវិភាគបម្រើបម្រាស់ទិន្នន័យនៃប្រព័ន្ធ.....	៥៥
៣.២ ផ្ទាំង Interface នៃប្រព័ន្ធ.....	៦០

សេចក្តីសន្និដ្ឋាន និងការផ្តល់អនុសាសន៍

១ សេចក្តីសន្និដ្ឋាន.....	៦៨
២ ការផ្តល់អនុសាសន៍.....	៦៨

ឯកសារយោង

ឧបសម្ព័ន្ធ

បញ្ជីអក្សរកាត់

RAD	: Rapid Application Development
ISP	: Internet Service Provider
URL	: Uniform Resource Locator
HTML	: Hyper Text Markup Language
E-commerce	: Electronic commerce
DNS	: Domain Name System Server
TLD	: Top Level Domains
FTP	: File Transfer Protocol
IDE	: Integrated Development Environment
CSS	: Cascading Style Sheet
CDN	: Content Distribution Network
PHP	: Hypertext Preprocessor
MVC	: Model View Controller
JSON	: JavaScript Object Notation
SSL	: Secure Sockets Layer
PDO	: PHP Data Objects
XML	: Extensible Markup Language
HTTP	: Hyper Text Transfer Protocol
DBMS	: Database Management System
SQL	: Structured Query Language

បញ្ជីរូបភាព

រូបភាពទី ១៖ Waterfall Model Diagram.....	៦
រូបភាពទី ២៖ ទំនាក់ទំនង Internet	៩
រូបភាពទី ៣៖ Static Website Process	១២
រូបភាពទី ៤៖ Dynamic Website Process	១២
រូបភាពទី ៥៖ Personal Website	១៣
រូបភាពទី ៦៖ Information Website	១៤
រូបភាពទី ៧៖ Career Website	១៥
រូបភាពទី ៨៖ E-commerce Website	១៥
រូបភាពទី ៩៖ Entertainment & News Website	១៦
រូបភាពទី ១០៖ គំរូរូប និង Logo សម្គាល់ HTML	១៨
រូបភាពទី ១១៖ គំរូរូប Internal និង External CSS).....	១៩
រូបភាពទី ១២៖ Bootstrap.....	២០
រូបភាពទី ១៣៖ របៀបទី ១ នៃការប្រើប្រាស់ bootstrap	២១
រូបភាពទី ១៤៖ របៀបទី ២ នៃការប្រើប្រាស់ bootstrap	២១
រូបភាពទី ១៥៖ តារាង Event JQuery	២៦
រូបភាពទី ១៦៖ ទម្រង់ MVC របស់ Laravel.....	២៧
រូបភាពទី ១៧៖ របៀប Laravel ធ្វើការ.....	២៨
រូបភាពទី ១៨៖ បង្ហាញពីទម្រង់របស់ Router	២៨
រូបភាពទី ១៩៖ ដំណើរការរបស់ Middleware.....	២៩
រូបភាពទី ២០៖ ទម្រង់ Middleware	២៩
រូបភាពទី ២១៖ បង្ហាញពី របៀប assign middleware key ទៅក្នុង Kernel.php.....	៣០
រូបភាពទី ២២៖ បង្ហាញពី ការប្រើ HTTP request	៣០
រូបភាពទី ២៣៖ បង្ហាញពី ការ validate តាម \$request validate.....	៣១
រូបភាពទី ២៤៖ បង្ហាញពី ការ validate តាម Validator instance	៣១
រូបភាពទី ២៥៖ បង្ហាញពី ទម្រង់របស់ View	៣២
រូបភាពទី ២៦៖ បង្ហាញពី ការ connection database ក្នុង .env file)	៣៣
រូបភាពទី ២៧៖ បង្ហាញពីទម្រង់របស់ Migration	៣៣
រូបភាពទី ២៨ ៖ តារាងបញ្ជាក់ Data Type របស់ PHP	៣៨

រូបភាពទី ២៩៖ តារាងពិពណ៌នា Arithmetic Operator	៣៩
រូបភាពទី ៣០៖ តារាងពិពណ៌នា Assignment Operator	៣៩
រូបភាពទី ៣១៖ តារាងពិពណ៌នា Comparison Operator	៤០
រូបភាពទី ៣២៖ តារាងពិពណ៌នា Comparison Operator	៤០
រូបភាពទី ៣៣៖ តារាងពិពណ៌នា Incrementing and Decrementing Operator	៤១
រូបភាពទី ៣៤៖ Flowchart If statement	៤១
រូបភាពទី ៣៥៖ Flowchart If-else-if statement	៤២
រូបភាពទី ៣៦៖ រូបភាពតំណាងឱ្យ MySQL.....	៤៨
រូបភាពទី ៣៧៖ Interface របស់ PHP MyAdmin.....	៤៩
រូបភាពទី ៣៨៖ តារាងពិពណ៌នា Database Relationship.....	៥០
រូបភាពទី ៣៩៖ តារាងគម្រោងពេលវេលានៃការស្រាវជ្រាវ.....	៥៤
រូបភាពទី ៤០ ៖ បង្ហាញពី Login Page.....	៦១
រូបភាពទី ៤១៖ បង្ហាញពី Home Page.....	៦២
រូបភាពទី ៤២ ៖ បង្ហាញពី Reservation Page	៦៣
រូបភាពទី ៤៣៖ បង្ហាញពី Reservation Detail	៦៣
រូបភាពទី ៤៤ ៖ បង្ហាញពី Form add Reservation.....	៦៤
រូបភាពទី ៤៥ ៖ បង្ហាញពី check-in page	៦៤
រូបភាពទី ៤៦ ៖ បង្ហាញពី form កែប្រែព័ត៌មាន Check-in	៦៥
រូបភាពទី ៤៧ ៖ បង្ហាញពីផ្ទាំង Check-Out	៦៥
រូបភាពទី ៤៨ ៖ បង្ហាញពីផ្ទាំង Payment Page.....	៦៦
រូបភាពទី ៤៩៖ បង្ហាញពីផ្ទាំង Form Payment.....	៦៦
រូបភាពទី ៥០៖ បង្ហាញពីផ្ទាំង Users	៦៧
រូបភាពទី ៥១ ៖ បង្ហាញពីផ្ទាំងបញ្ចូលព័ត៌មានអ្នកប្រើប្រាស់.....	៦៧
រូបភាពទី ៥២ ៖ បង្ហាញពីផ្ទាំងបញ្ចូលកែប្រែព័ត៌មានអ្នកប្រើប្រាស់.....	៦៨

សេចក្តីផ្តើម

១. លំនាំបញ្ជាក់នៃការស្រាវជ្រាវ

ក្នុងពេលបច្ចុប្បន្ននេះ វិស័យបច្ចេកវិទ្យាមានភាពរីកចម្រើនយ៉ាងខ្លាំង ពីមួយថ្ងៃទៅមួយថ្ងៃ ហើយគេបានយកវាទៅប្រើប្រាស់ ពាសពេញសកលលោកផងដែរ។ មិនត្រឹមតែប៉ុណ្ណោះវិស័យ ព័ត៌មានវិទ្យាក៏បានដើរតួនាទីយ៉ាងសំខាន់ក្នុងការអភិវឌ្ឍសេដ្ឋកិច្ចរបស់ប្រទេសជាតិផងដែរ។ លើសពីនេះទៅទៀតគេបានចាត់ទុកវិស័យនេះជាវិស័យចម្បងក្នុងការជម្រុញអភិវឌ្ឍន៍ និងសេវាកម្ម ឲ្យកាន់តែរីកចម្រើនផងដែរ។ ជាក់ស្តែងបើយើងក្រឡេកទៅមើលបណ្តាប្រទេសដែលមានការរីក ចម្រើនខ្លាំងដូចជាប្រទេស សហរដ្ឋអាមេរិច រុស្ស៊ី ជប៉ុន កូរ៉េ ចិន និងប្រទេសដែលនៅសមាគមន៍ អឺរ៉ុបជាដើមបានយកវិស័យព័ត៌មានវិទ្យា កំណត់ថាជាវិស័យសំខាន់ក្នុងការសម្រួលកិច្ចការដឹកនាំ ប្រទេសជាតិរបស់ខ្លួន ។

ដោយឡែកយើងក្រឡេកមកមើលប្រទេសកម្ពុជាយើងវិញ វិស័យព័ត៌មានវិទ្យាក៏កំពុងតែ មានការរីកចម្រើនផងដែរដែលធ្វើឲ្យមានការអភិវឌ្ឍយ៉ាងឆាប់រហ័ស ទាំងផ្នែកសេដ្ឋកិច្ច សង្គមកិច្ច និងស្ទើរតែគ្រប់វិស័យ។ ជាក់ស្តែងណាស់នៅក្នុងការគ្រប់គ្រងរបស់ស្ថាប័នរដ្ឋ និងឯកជនមួយចំនួន បានចាប់ផ្តើមផ្លាស់ប្តូរពីការគ្រប់គ្រងទិន្នន័យដោយការសរសេរដៃ ដែលមានលក្ខណៈពិបាក គ្រប់គ្រងទៅជាការគ្រប់គ្រងដោយបច្ចេកវិទ្យាព័ត៌មានវិទ្យាដើម្បីបង្កើនប្រសិទ្ធភាព និងផលិតកម្មឱ្យ បានគ្រប់គ្រាន់ដើម្បីឆ្លើយតបទៅនឹងតម្រូវការទីផ្សារ និងការប្រកួតប្រជែងសេដ្ឋកិច្ចទីផ្សារសេរីនៅ ពេលបច្ចុប្បន្ន។ ម្យ៉ាងវិញទៀត យើងសង្កេតឃើញថាក្នុងប្រទេសកម្ពុជាយើងមានការរីកចម្រើន យ៉ាងខ្លាំងផ្នែកទេសចរណ៍ ទាំងក្នុងស្រុក និងក្រៅស្រុកដែលជាហេតុធ្វើឲ្យមានការរីកដុះដាលយ៉ាង ច្រើននូវសំណង់អាគារដូចជា សណ្ឋាគារ ផ្ទះសំណាក់ នៅតាមតំបន់ទេសចរណ៍ទាំងអស់នោះ។ មិនត្រឹមតែប៉ុណ្ណោះ នៅក្នុងរាជធានីភ្នំពេញ និងតាមបណ្តាខេត្តផ្សេងៗនៅក្នុងប្រទេសកម្ពុជា ក៏ សម្បូរទៅដោយ សណ្ឋាគារ និងផ្ទះសំណាក់ផងដែរ ដែលសណ្ឋាគារ និងផ្ទះសំណាក់ទាំងអស់នោះ សុទ្ធសឹងតែត្រូវការវិស័យព័ត៌មានវិទ្យាដើម្បីជំរុញឲ្យមានភាពងាយស្រួលក្នុងការគ្រប់គ្រងទិន្នន័យ និងមានភាពឆាប់រហ័ស ។

២. មូលដ្ឋាននៃការស្រាវជ្រាវ

ឆ្លងតាមការសិក្សាអស់រយៈពេលប៉ុន្មានឆ្នាំកន្លងមកនេះ យើងខ្ញុំបានទទួលចំណេះដឹងភាគច្រើនទាក់ទងនឹងទ្រឹស្តី និងការសរសេរកូដ ប៉ុន្តែមិនទាន់បានអនុវត្តនិងស្រាវជ្រាវឲ្យស៊ីជម្រៅនៅឡើយទេ។ ដូច្នេះហើយយើងខ្ញុំក៏សូមអរគុណដល់សាកលវិទ្យាល័យដែលបានអនុញ្ញាតឲ្យយើងខ្ញុំសរសេរសារណាបញ្ចប់ការសិក្សា ដើម្បីបង្កើនសមត្ថភាពបន្ថែមទៀតផ្នែកព័ត៌មានវិទ្យា។ យើងខ្ញុំបានជ្រើសរើសការបង្កើត **កម្មវិធីប្រព័ន្ធគ្រប់គ្រងសណ្ឋាគារ** មកធ្វើការសិក្សាស្រាវជ្រាវ។

៣. ចំណោទបញ្ហានៃការស្រាវជ្រាវ

តាមរយៈការសិក្សាស្រាវជ្រាវ បានបង្ហាញឲ្យឃើញថា នៅតាមសណ្ឋាគារ និងផ្ទះសំណាក់មួយចំនួនគឺគ្មានការគ្រប់គ្រងព័ត៌មានជាលក្ខណៈបច្ចេកវិទ្យាឡើយ ពោលគឺគ្រប់គ្រងមានលក្ខណៈកត្រាព័ត៌មានដោយដៃនៅលើសៀវភៅ និងប្រើប្រាស់កម្មវិធីដែលមិនបានចងក្រងជាប្រព័ន្ធដូចជា Microsoft Excel ជាដើម។ មូលហេតុទាំងអស់នេះហើយទើបធ្វើឲ្យការគ្រប់គ្រង សណ្ឋាគារ ឬ ផ្ទះសំណាក់របស់ពួកគេមានការលំបាក និងជួបប្រទះនូវបញ្ហាដូចខាងក្រោម៖

- ការគ្រប់គ្រងព័ត៌មានរបស់អតិថិជនមានលក្ខណៈស្មុគស្មាញ ។
- ពិបាកស្វែងរកទិន្នន័យ និងព័ត៌មានរបស់អតិថិជន ។
- ការគ្រប់គ្រងមានលក្ខណៈយឺតយ៉ាវ ។
- មិនមានការស្វែងរករបាយការណ៍តាម ថ្ងៃ ខែ ។
- ទិន្នន័យមិនមានសុវត្ថិភាព ។

បញ្ហាដែលបានលើកឡើងខាងលើនឹងត្រូវបានដោះស្រាយតាមរយៈការបង្កើតប្រព័ន្ធគ្រប់គ្រងសណ្ឋាគារនេះ។

៤. គោលបំណងនៃការស្រាវជ្រាវ

ការស្រាវជ្រាវមែងមានគោលបំណងដើម្បីជាផ្លូវក្នុងការកំណត់ឲ្យបានច្បាស់លាស់ សមស្របនឹងប្រធានបទ ។ ដើម្បីដោះស្រាយជូនម្ចាស់ផ្ទះសំណាក់ ឬ សណ្ឋាគារ ចំពោះបញ្ហាដែលកំពុងជួបប្រទះ ក៏ដូចជាបង្កើននូវគុណភាព និងជួយដល់សកម្មភាពគ្រប់គ្រងរបស់ពួកគាត់ឲ្យកាន់តែមានប្រសិទ្ធភាព និងភាពងាយស្រួល ការសិក្សាស្រាវជ្រាវមួយនេះ ត្រូវបានធ្វើឡើងក្នុងគោលបំណងបង្កើតឲ្យបានកម្មវិធីគ្រប់គ្រងសណ្ឋាគារមួយដែលមានគោលដៅសំខាន់ៗដូចខាងក្រោម៖

- ផ្តល់លទ្ធភាពដល់អ្នកគ្រប់គ្រងអាចធ្វើការបញ្ចូលទិន្នន័យ រក្សាទុកទិន្នន័យ ធ្វើការកែប្រែទិន្នន័យ និង បោះទាញយកទិន្នន័យពី Database មកធ្វើការវិភាគ ។
- ផ្តល់លទ្ធភាពដល់អ្នកប្រើប្រាស់ធ្វើការគ្រប់គ្រងលើសុវត្ថិភាពទិន្នន័យ ។

- ផ្តល់លទ្ធភាពដល់អ្នកប្រើប្រាស់នៃការស្វែងរកទិន្នន័យផ្សេងៗបានយ៉ាងឆាប់រហ័សដោយក្រោយរយៈពេលខ្លី ។
- កាត់បន្ថយភាពស្មុគស្មាញក្នុងប្រតិបត្តិការគ្រប់គ្រងពីមុន ។
- អនុញ្ញាតឱ្យ Admin អាចតាមដាននូវសកម្មភាពរបស់អ្នកប្រើប្រាស់ប្រព័ន្ធទាក់ទងនឹង Check-In ឬ Check-Out ។

៥. វិសាលភាព និង ដែនកំណត់នៃការស្រាវជ្រាវ

ការបង្កើតកម្មវិធីគ្រប់គ្រងសណ្ឋាគារនេះគឺមានលក្ខណៈជា WEB APPLICATION ដូច្នេះហើយវាដំណើរការនៅលើ Web Browser។ ដើម្បីបង្កើតកម្មវិធីមួយឲ្យមានគុណភាពល្អ សមស្របជាមួយនឹងពេលវេលា និង សមត្ថភាព យើងត្រូវកំណត់នូវដែនកំណត់របស់វាឲ្យបាន ច្បាស់លាស់ ។ ខាងក្រោមនេះជាដែនកំណត់ និងវិសាលភាពនៃការសិក្សាស្រាវជ្រាវ៖

- អាចកំណត់សិទ្ធិរបស់អ្នកប្រើប្រាស់ក្នុងការប្រើប្រាស់កម្មវិធីទៅតាមក្រុមដែលបានកំណត់
- អាចគ្រប់គ្រងព័ត៌មានរបស់អតិថិជន
- អាចធ្វើការកក់បន្ទប់មុនពេលចូលស្នាក់នៅ
- អាចធ្វើការគ្រប់គ្រងការ Check-in និង Check-out របស់អតិថិជន
- អាចត្រួតពិនិត្យចំនួនបន្ទប់នៅក្នុងសណ្ឋាគារ
- អាចធ្វើការទូទាត់ទឹកប្រាក់ដែលអតិថិជនចូលស្នាក់នៅ
- អាចត្រួតពិនិត្យចំណូល និងចំណាយរបស់សណ្ឋាគារ
- អាចត្រួតពិនិត្យ និង ធ្វើរបាយការណ៍

៦. អត្ថប្រយោជន៍នៃការសិក្សាស្រាវជ្រាវ

ការសិក្សាស្រាវជ្រាវទៅលើការបង្កើតប្រព័ន្ធគ្រប់គ្រងការលក់នេះផ្តល់នូវអត្ថប្រយោជន៍ចំនួន ៣ សំខាន់ៗគឺ៖

- ចំពោះអ្នកសិក្សា
- ចំពោះអ្នកគ្រប់គ្រងអជីវកម្មសណ្ឋាគារ ឬ ផ្ទះសំណាក់
- ចំពោះសង្គមជាតិ

ក. ចំពោះអ្នកសិក្សា

- ទទួលបាននូវបទពិសោធន៍និងចំណេះដឹងថ្មីតាមរយៈការសិក្សាស្រាវជ្រាវទាក់ទងនឹងការបង្កើតប្រព័ន្ធគ្រប់គ្រងសណ្ឋាគារ។

- ផ្តល់ឱកាសឲ្យក្រុមយើងខ្ញុំបានអនុវត្តផ្ទាល់នូវទ្រឹស្តីដែលបានសិក្សាអស់រយៈពេល 4 ឆ្នាំ កន្លងមក។

- អាចឲ្យអ្នកដឹងពីចំណុចខ្លាំងនិងចំណុចខ្វះខាតក្នុងការសិក្សាស្រាវជ្រាវលើការអនុវត្តជាក់ស្តែងលើការបង្កើតប្រព័ន្ធនេះ។

ខ. ចំពោះអ្នកគ្រប់គ្រងអជីវកម្មសណ្ឋាគារ ឬ ផ្ទះសំណាក់

- អាចប្រើប្រាស់នូវប្រព័ន្ធគ្រប់គ្រងដើម្បីផ្តល់សេវាកម្ម និងបង្កើនល្បឿនការងារលទ្ធភាព ផលលិតកម្ម និង ប្រសិទ្ធភាពបានយ៉ាងប្រសើរ។

- ធ្វើឲ្យការលក់ដូររបស់ពួកគេមានសុវត្ថិភាពទិន្នន័យរៀបរាប់សាងការលូចនិងបាត់បង់ដោយប្រការណាមួយ។

- ទិន្នន័យត្រូវបានរក្សាទុកនៅក្នុង Database អ្នកប្រើប្រាស់អាចត្រួតពិនិត្យយកមកធ្វើរបាយការណ៍បានយ៉ាងងាយស្រួល។

- ងាយស្រួលក្នុងការគ្រប់គ្រងចំណូលចំណាយរបស់សណ្ឋាគារ ឬ ផ្ទះសំណាក់។

គ. ចំពោះសង្គមជាតិ

- ផ្តល់ឯកសារស្រាវជ្រាវសម្រាប់អ្នកស្រាវជ្រាវទូទៅពាក់ព័ន្ធនឹង ប្រព័ន្ធគ្រប់គ្រងសណ្ឋាគារ ឬ ផ្ទះសំណាក់។

- ជម្រុញឲ្យមានការសិក្សាស្រាវជ្រាវបន្ថែមទៀតដើម្បីអភិវឌ្ឍន៍សង្គមជាតិទៅតាមសម័យកាលដែលជឿនលឿនមួយនេះ។

- ផ្តល់ជាគំរូដ៏ល្អដល់អ្នកសិក្សាជំនាន់ក្រោយក្នុងការសិក្សាស្រាវជ្រាវដោយខ្លួនឯង

- រួមចំណែកក្នុងការអភិវឌ្ឍន៍សមត្ថភាពធនធានមនុស្សនៅក្នុងប្រទេសយើង។

ជំពូកទី ១

ទ្រឹស្តីពាក់ព័ន្ធនឹងការបង្កើតប្រព័ន្ធ

ដើម្បីមានភាពងាយស្រួលក្នុងការធ្វើសារណា និងបង្កើតជាប្រព័ន្ធកម្មវិធីគ្រប់គ្រងសណ្ឋាគារ ឲ្យមានដំណើរការបានល្អក្រុមរបស់យើងខ្ញុំបានស្រាវជ្រាវ និងចងក្រងទ្រឹស្តីដែលពាក់ព័ន្ធមួយចំនួន ក្នុងគោលបំណងដើម្បីជួយដោះស្រាយបញ្ហាក្នុងការបង្កើតប្រព័ន្ធគ្រប់គ្រងរបស់យើង។

១.១ ទ្រឹស្តីពាក់ព័ន្ធនឹង Software Engineering

១.១.១ និយមន័យ Software Engineering

Software Engineering គឺជាដំណើរការនៃការបង្កើត software មួយដែលរួមមានការធ្វើ គម្រោង ឬផែនការ(Planning) ការចនាប្រព័ន្ធ(Design) ការអនុវត្ត ឬការសរសេរកូដ (Implementation or Coding) ការធ្វើតេស្ត(Testing) ការប្រើប្រាស់ និងការថែរក្សា(Maintenance)។ ការបង្កើតប្រព័ន្ធ Software ត្រូវប្រកបដោយគុណភាព សុវត្ថិភាព និងមានការជឿជាក់ពីសំណាក់អ្នក ប្រើប្រាស់ប្រព័ន្ធ។

១.១.២ ម៉ូដែល (Model)

The software development models ជាដំណើរការឬវិធីសាស្ត្រផ្សេងៗដែលត្រូវបានជ្រើស រើសសម្រាប់ការអភិវឌ្ឍគម្រោងអាស្រ័យលើគោលបំណង និងគោលដៅរបស់គម្រោង។មានម៉ូ- ដែលជាច្រើនដែលត្រូវបានបង្កើតឡើងដើម្បីសម្រេចគោលបំណងផ្សេងៗគ្នា។ ការជ្រើសរើសម៉ូ- ដែលត្រឹមត្រូវសម្រាប់ការអភិវឌ្ឍកម្មវិធីគឺសំខាន់ណាស់។ ខាងក្រោមនេះជាម៉ូដែលមួយចំនួនដែល គេតែងតែយកទៅប្រើប្រាស់ក្នុងប្រព័ន្ធ ៖

- Waterfall Model
- V Model
- Incremental Model
- RAD Model
- Agile Model
- Spiral Model
- Prototype Model

១.១.៣ អ៊ីទៅវា Waterfall Model

Waterfall Model គឺជាម៉ូដែលដំបូងគេដែលត្រូវបានណែនាំ។ ម៉ូដែលនេះមានលក្ខណៈសាមញ្ញ ងាយយល់ និងងាយប្រើប្រាស់។ វាចែកចេញជា ៦ ដំណាក់កាល ហើយដំណាក់កាលនីមួយៗត្រូវតែធ្វើឲ្យរួចរាល់ មុនពេលធ្វើដំណាក់កាលបន្ទាប់។ ជាធម្មតាគេតែងតែប្រើប្រាស់វាសម្រាប់គម្រោងដែលតូចៗ និងមានតម្រូវការមិនច្បាស់លាស់។ ខាងក្រោមនេះនេះជាដំណាក់កាលនីមួយៗរបស់គំរូ Waterfall Model ៖

ដំណាក់កាលទី ១៖ Requirement Analysis

ដំណាក់កាលទី ២៖ System Design

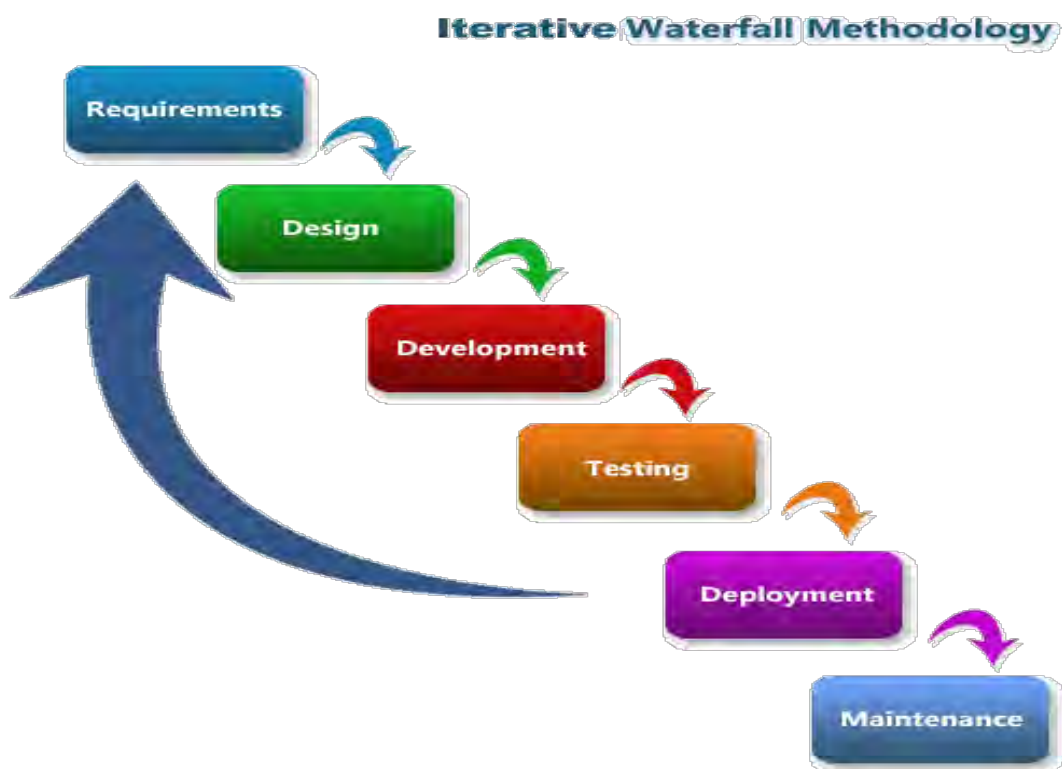
ដំណាក់កាលទី ៣៖ Implementation

ដំណាក់កាលទី ៤៖ Testing

ដំណាក់កាលទី ៥៖ Development

ដំណាក់កាលទី ៦៖ Maintenance

ដំណាក់កាលទាំងស្រុងសម្រាប់សុទ្ធតែមានមុខងារ និងដំណើរការរៀងៗខ្លួន។



(រូបភាពទី ១៖ Waterfall Model Diagram)

១.១.៣.១ Requirement Gathering Analysis

ដំណាក់កាលនេះពាក់ព័ន្ធនឹងគោលបំណងក្នុងការបង្កើតប្រព័ន្ធ និងការប្រមូលទិន្នន័យពីអតិថិជន ឬ អ្នកប្រើប្រាស់ដើម្បីទទួលបានទិន្នន័យ និងដឹងពីផលពិបាកដែលពួកគេជួបប្រទះហើយស្វែងយល់ពីតម្រូវការច្បាស់លាស់ពីអ្នកប្រើប្រាស់ ដើម្បីងាយស្រួលក្នុងការកំណត់មុខងារនៅក្នុងប្រព័ន្ធកម្មវិធី ។ បន្ទាប់ពីធ្វើការប្រមូលទិន្នន័យបានគ្រប់គ្រាន់ហើយ ទើបយកទិន្នន័យទាំងអស់នោះមកធ្វើការវិភាគ ។ ការវិភាគគឺត្រូវបានបែងចែកចេញជាពីរគឺ ការវិភាគទៅលើតម្រូវការរបស់អ្នកប្រើប្រាស់កម្មវិធីមួយទៀតគឺវិភាគទៅលើទិន្នន័យរបស់កម្មវិធី។

ក.ការវិភាគទៅលើតម្រូវការអ្នកប្រើប្រាស់

ការវិភាគនេះគឺសំដៅទៅលើការវិភាគ Interface និងមុខងារសំខាន់ៗសម្រាប់អតិថិជនប្រើប្រាស់នៅក្នុងកម្មវិធី ទៅតាមតម្រូវការដែលអ្នកប្រើប្រាស់ចង់បាន ។

ខ.ការវិភាគ Database

គឺជាការវិភាគទៅលើកន្លែងផ្ទុកទិន្នន័យនិងប្រភេទទិន្នន័យផ្សេងៗ ដើម្បីរៀបចំទិន្នន័យទៅតាមប្រភេទទិន្នន័យនីមួយៗនៅក្នុងតារាង(Table) ជាពិសេសគឺការវិភាគទៅលើទំនាក់ទំនងទិន្នន័យ (Relationship) ។

១.១.៣.២ System Design

បន្ទាប់ពីធ្វើការវិភាគហើយទាំងផ្នែក Interface និង ផ្នែក Database យើងធ្វើការ Design Interface ឲ្យសមស្របទៅនឹងតម្រូវការអ្នកប្រើប្រាស់ និង មើលទៅមានភាពងាយស្រួលដល់អ្នកប្រាស់ ហើយមានលក្ខណៈសមរម្យ ។

១.១.៣.៣ Implementation

ក្នុងដំណាក់កាលនេះគឺយើងធ្វើការសរសេរកូដដើម្បីធ្វើការបញ្ជាទៅលើ Interface ដែលយើងបាន Design នោះឲ្យធ្វើការទៅតាមអ្វីដែលយើងចង់បាន និងសរសេរកូដដាក់លក្ខខណ្ឌមួយចំនួនផងដែរ ដែលយើងអាចនិយាយបានថាគឺសរសេរកូដដើម្បីគ្រប់គ្រង Interface ទាំងមូលតែម្តង។

១.១.៣.៤ Testing

ក្រោយពីប្រព័ន្ធទាំងមូលត្រូវបានបង្កើតហើយជំហានបន្ទាប់គឺការសាកល្បងប្រព័ន្ធ។ ដំណាក់កាលនេះតម្រូវឲ្យអ្នកបង្កើតកម្មវិធីធ្វើការតេស្តទៅលើប្រព័ន្ធ ដើម្បីស្វែងរកកំហុសឆ្គងផ្សេងៗដែលកើតមានក្នុងដំណើរការរបស់វា និងពិនិត្យមើលការពេញចិត្តរបស់អ្នកប្រើប្រាស់ ។ បើសិនជាមានបញ្ហាណាមួយឬការវិភាគតម្រូវការរបស់ប្រព័ន្ធមិនបានគ្រប់ជ្រុងជ្រោយត្រូវ យើងត្រូវធ្វើការវិភាគម្តងទៀត ។

១.១.៣.៥ Development of System

ដំណាក់កាលនេះយើងត្រូវយកប្រព័ន្ធដើម្បីឲ្យអតិថិជនដើម្បីប្រើប្រាស់។

១.១.៣.៦ Maintenance

នេះជាដំណាក់កាលចុងក្រោយរបស់ Waterfall Model ដែលដំណាក់កាលនេះជាដំណាក់កាលថែទាំកម្មវិធី និងកែប្រែបញ្ហាឬកំហុសឆ្គងមួយចំនួននៅពេលអតិថិជនមិនពេញចិត្ត ឬ បញ្ហាលើប្រព័ន្ធក្នុងកំឡុងពេលអតិថិជនយកទៅប្រើប្រាស់ ។

❖ **គុណសម្បត្តិនិងគុណវិបត្តិរបស់ water fall model**

➤ **គុណសម្បត្តិ**

- ងាយស្រួលក្នុងការប្រើប្រាស់ និងងាយស្រួលយល់
- ងាយស្រួលក្នុងការគ្រប់គ្រង និងត្រួតពិនិត្យឡើងវិញ
- ដំណើរការល្អជាមួយ Project តូចៗ
- មានការបែងចែកដំណាក់កាលច្បាស់លាស់
- ជំហាននៃការ Processes និង Complete ក្នុងពេលតែមួយ
- ងាយស្រួលក្នុងការរៀបចំការងារ
- ដំណើរការ និង លទ្ធផលមានការចងក្រងបានយ៉ាងល្អ

➤ **គុណវិបត្តិ**

- ហានិភ័យខ្ពស់
- មានផលមិនល្អសម្រាប់ Project ដែលមានរយៈពេលវែង និង ធំៗ
- ប្រព័ន្ធដែលបានបង្កើតអាចដំណើរការបាន លុះត្រាតែបំពេញដំណាក់កាលទាំងអស់ឲ្យបានចប់សព្វគ្រប់ និងត្រឹមត្រូវ
- មិនល្អសម្រាប់ Project ដែលស្មុគស្មាញ
- ការកែប្រែដែនកំណត់ក្នុងពេលដំណើរការអាចធ្វើឲ្យគម្រោងប្រឈមមុខនឹងហានិភ័យ ឬ អាចឈានដល់ការបញ្ចប់។

១.២ ទ្រឹស្តីពាក់ព័ន្ធនឹងអ៊ីនធឺណេត

១.២.១ ប្រភេទស្ថានភាពនៃអ៊ីនធឺណេត

មនុស្សជាច្រើន នាពេលបច្ចុប្បន្ននេះ កំពុងប្រើអ៊ីនធឺណេត សំរាប់ការទាក់ទងគ្នាតាម ឆាត ឬ តាមវីធីផ្សេងៗទៀត និងមានបណ្តាញទំនាក់ទំនងសង្គមជាច្រើនតាមអ៊ីនធឺណេត ដែលឥលូវនេះមានសារៈសំខាន់បំផុត។

អ៊ីនធឺណែតចាប់ផ្តើម នៅទសវត្សរ៍ទី៦០ សតវត្សរ៍ទី២០ នៅ សហរដ្ឋអាមេរិក នៅពេលរដ្ឋាភិបាល និងក្រុមហ៊ុនខ្លះ ចង់បង្កើតបណ្តាញកុំព្យូទ័រ ដែលជួយដំណើរការក្នុង ការទាក់ទង។ នៅទសវត្សរ៍ទី៨០ សតវត្សរ៍ទី២០ «មូលស្ថាបនាជាតិ របស់វិទ្យាសាស្ត្រ នៃសហរដ្ឋអាមេរិក» (Science National Foundation) បាននាំបង្កើតអ៊ីនធឺណែតជាមួយ ក្រុមហ៊ុនមួយផ្សេងទៀត សំរាប់ប្រើវា នៅលើពិភពលោកនាទសវត្សរ៍ទី៩០ សតវត្សរ៍ទី២០ អ៊ីនធឺណែតបានក្លាយទៅជាជំនួញ និងចាប់ផ្តើមកើតមាន «បដិវត្តន៍ នៃបច្ចេកវិជ្ជាលើកទី៣» ។

១.២.២ អ្វីទៅជាអ៊ីនធឺណែត

អ៊ីនធឺណែត រឺ ភាសាខ្មែរថា អន្តរបណ្តាញ គឺជាការភ្ជាប់ ឬ ការធ្វើអោយមានទំនាក់ទំនងគ្នា រវាងកុំព្យូទ័រជាច្រើននៅជុំវិញពិភពលោកក្នុងគោលបំណងប្រាស្រ័យទាក់ទងគ្នា និងចែកចាយទិន្នន័យ។ អ៊ីនធឺណែតបែងចែកជាពីរ មានខ្សែ និង ឥតខ្សែ ។ ទោះបីជាអ៊ីនធឺណិតមានភាពទូលំទូលាយយ៉ាងណាក៏ដោយ ប៉ុន្តែអ្នកប្រើប្រាស់ភាគច្រើនទើបតែដឹងពីអត្ថប្រយោជន៍នៃការប្រើប្រាស់វា ហើយទទឹមនិងនេះវាក៏មានផលប៉ះពាល់ដល់អ្នកប្រើប្រាស់មួយចំនួនបងដែរ។



(រូបភាពទី ២៖ ទំនាក់ទំនង Internet)

ខាងក្រោមនេះជាផលប៉ះពាល់ និងផលប្រយោជន៍របស់អ៊ីនធឺណិត៖

ក. ផលប្រយោជន៍

- **ការទំនាក់ទំនងលឿនជាងមុន** មានន័យថា ទំនាក់ទំនងកាន់តែឆាប់រហ័ស អាចឲ្យយើងធ្វើឯកសារឬលិខិតផ្សេងៗ ទៅកាន់មនុស្សនៅជុំវិញខ្លួនយើង តាមរយៈសារអេឡិចត្រូនិច Email ក្នុងល្បឿនយ៉ាងឆាប់រហ័ស ទោះបីស្ថិតនៅចម្ងាយឆ្ងាយឆ្ងាយក៏ដោយ ក៏អាចបញ្ជូនទៅឲ្យក្នុងរយៈពេលមួយប៉ប្រិចភ្នែកប៉ុណ្ណោះ។

- **ធនធានព័ត៌មានច្រើនក្រៃលែង** ប្រភពព័ត៌មានសម្បូរបែប មានន័យថា នៅលើ អ៊ីនធឺណែតមានប្រមូលផ្តុំទៅដោយចំណេះដឹងគ្រប់រូបភាព គ្រប់វិស័យទាំងអស់ ជាពិសេសចំណេះដឹងអប់រំតែម្តង។

- **ផ្តល់ការអប់រំមិនចេះអស់** ងាយស្រួលរកឯកសារសិក្សាតាមបណ្តាញ អ៊ីនធឺណិត។

- **ផ្តល់សេវាកម្មលើបណ្តាញអ៊ីនធឺណិត និងពាណិជ្ជកម្ម** សំដៅទៅលើការធ្វើ Business តាមអនឡាញ។ យើងអាចផ្សព្វផ្សាយពាណិជ្ជកម្មអ្វីមួយនៅលើវាបាន ហើយថែមទាំងអាចធ្វើការទិញដូរ នៅលើអនឡាញបានទៀតផង។

- **បណ្តាញទំនាក់ទំនងសង្គមបណ្តាញទំនាក់ទំនងសង្គម** គឺ យើងអាចមានសកម្មភាពក្នុងសង្គមតាមរយៈការប្រឡូកលេង កម្មវិធីបណ្តាញសង្គមមួយចំនួនដូចជា Facebook, Twitter, Google Plus, Pinterest ,Skype និង Instagram ជាដើម។

ខ. ផលប៉ះពាល់

- **លួចព័ត៌មានផ្ទាល់ខ្លួន** សំដៅទៅលើការចូលលួចព័ត៌មានផ្ទាល់ខ្លួនរបស់អ្នកប្រើប្រាស់នៅពេល Access ចូលទៅកាន់ អ៊ីនធឺណែត ពីសំណាក់ក្រុមបិទមុខ Hacker។

- **មេរោគ** គឺការជ្រៀតចូលនូវ (មេរោគនៅ ពេលមានការផ្ញើ Email ឬធ្វើអ្វីមួយនៅលើអ៊ីនធឺណែត។

- **ប៉ះពាល់ដល់អ្នកមិនទាន់គ្រប់អាយុ** មានរឿងរ៉ាវខ្លះ វាប៉ះពាល់ដល់ស្មារតីអ្នកប្រើប្រាស់មិនទាន់គ្រប់អាយុ ដូចជាការចូលទៅបើកមើលវេបសាយ អាសអាភាស លេងហ្គេមអនឡាញ ជាដើម ដែលនេះជាគ្រោះថ្នាក់ធំបំផុត គួរជៀសវាងឲ្យឆ្ងាយសម្រាប់កុមារ។

- **ប៉ះពាល់សុខភាពខួរក្បាល**

១.២.៣ អ្វីទៅជា ISP

ISP គឺជាក្រុមហ៊ុនដែលផ្តល់សេវាកម្ម internet សំរាប់អោយអតិថិជនអាចដំណើរការ internet បាន។ នៅពេលប្រើប្រាស់ យើងនឹងconnectទៅកាន់ ISP បន្ទាប់មកក្រុមហ៊ុន ISP នឹងភ្ជាប់

Internet សំរាប់យើងប្រើប្រាស់។ នៅក្នុងប្រទេសយើងមានក្រុមហ៊ុន ISP ជាច្រើនដូចជាក្រុមហ៊ុន EZECOM, MekongNET, OpenNet, Online ជាដើម។

១.៣ ទ្រឹស្តីពាក់ព័ន្ធនៅលើគេហទំព័រ

១.៣.១ អ្វីទៅជាគេហទំព័រ

គេហទំព័រគឺជាបណ្តុំនៃ Web Page ជាច្រើនបញ្ចូលគ្នា បង្កើតបានជាទំនាក់ទំនងគ្នា ដែលមានរូបភាព វីដេអូ ព័ត៌មាន ចំនេះដឹង ឬអត្ថបទដែលអ្នកប្រើប្រាស់អាចចូលទៅកាន់គេហទំព័រទាំងអស់នោះតាមរយៈ URL ។ ភាគច្រើននៅតាមក្រុមហ៊ុន ឬ ស្ថាប័ននានាប្រើប្រាស់គេហទំព័រដើម្បីបង្ហាញព័ត៌មានរបស់ក្រុមហ៊ុន ឬស្ថាប័នរបស់ពួកគេ។

១.៣.២ អ្វីទៅជា Web Page

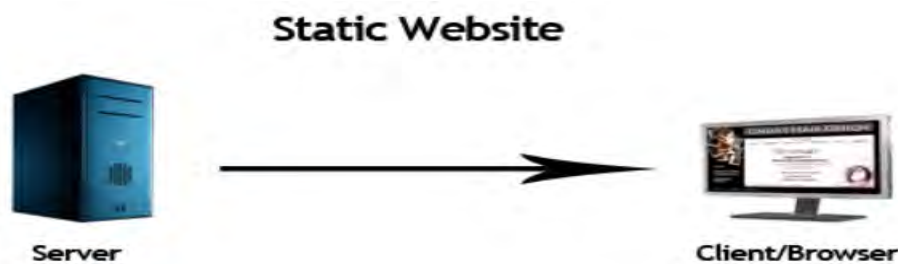
Web Page គឺជា page នីមួយៗរបស់គេហទំព័រដែលបង្ហាញទិន្នន័យ ឬព័ត៌មានដូចជា អត្ថបទ រូបភាព ឬវីដេអូ ជាដើម។ គេបង្ហាញ Web Page តាមរយៈ Web Browser ។ គេអាចបង្កើត Web Page តាមរយៈមធ្យោបាយជាច្រើនដោយពិភាក្សាលើភាសាដែលយកមកប្រើដូចជា ភាសាក្នុង Server Side Script និង Client Side Script ។

១.៣.៣ ប្រភេទនៃគេហទំព័រ

នៅក្នុងគេហទំព័រត្រូវបានគេដឹងថាបានបែងចែកចេញជាពីរប្រភេទគឺ Static និង Dynamic ។

១.៣.៣.១ Static Website

Static Website គឺជាវេបសាយមានលក្ខណៈសាមញ្ញ និងងាយស្រួលក្នុងការបង្កើត។ វាផ្ទុក Web Page ដែលមានលក្ខណៈថេរ។ ក្នុង page នីមួយៗគឺប្រើប្រាស់ HTML ដើម្បីបង្កើត និងបង្ហាញព័ត៌មានទៅ Web Page ដោយផ្ទាល់តែម្តង។ វេបសាយប្រភេទនេះគឺមិនទាមទារ Database ដើម្បីផ្ទុកទិន្នន័យឡើយ មានន័យថាព័ត៌មានទិន្នន័យដែលបង្ហាញលើ Web Page គឺសរសេរក្នុង tag HTML ផ្ទាល់តែម្តង។ ម្យ៉ាងវិញទៀតដើម្បីអាចកែប្រែទិន្នន័យបាន គឺយើងត្រូវចូលទៅកាន់ HTML កូដដើម្បីកែប្រែទិន្នន័យ ដែលធ្វើមានការពិបាកក្នុងការកែប្រែ។ Static Website ភាគច្រើនគេប្រើសម្រាប់បង្កើត Website ណាដែលមិនមានតម្រូវការក្នុងការផ្លាស់ប្តូរទិន្នន័យ ឬព័ត៌មានរបស់គេឡើយ ។



(រូបភាពទី ៣៖ Static Website Process)

១.៣.៣.២ Dynamic Website

Dynamic Website គឺជាវេបសាយដែលគេប្រើប្រាស់ ភាសា Programming ដើម្បីបង្កើតឡើង ហើយវាអាចបត់បែនបានតាមអ្វីដែលអ្នកប្រើប្រាស់ត្រូវការ។ រាល់ទិន្នន័យទាំងអស់គឺត្រូវបានរក្សា ទុកនៅក្នុង Database។ ទិន្នន័យដែលបញ្ចូល កែប្រែ និង បង្ហាញមកវិញគឺមានទំនាក់ទំនងជាមួយ Database ទាំងអស់។ ម្យ៉ាងវិញទៀតវាផ្តល់ភាពងាយស្រួលមែនទែនក្នុងការគ្រប់គ្រងដោយយើង អាចធ្វើការ បញ្ចូល កែប្រែ និងលុបទិន្នន័យនៅលើ Web Page ផ្ទាល់តែម្តង ។



(

រូបភាពទី ៥៖ Dynamic Website Process)

១.៣.៤ លក្ខណៈនៃគេហទំព័រ

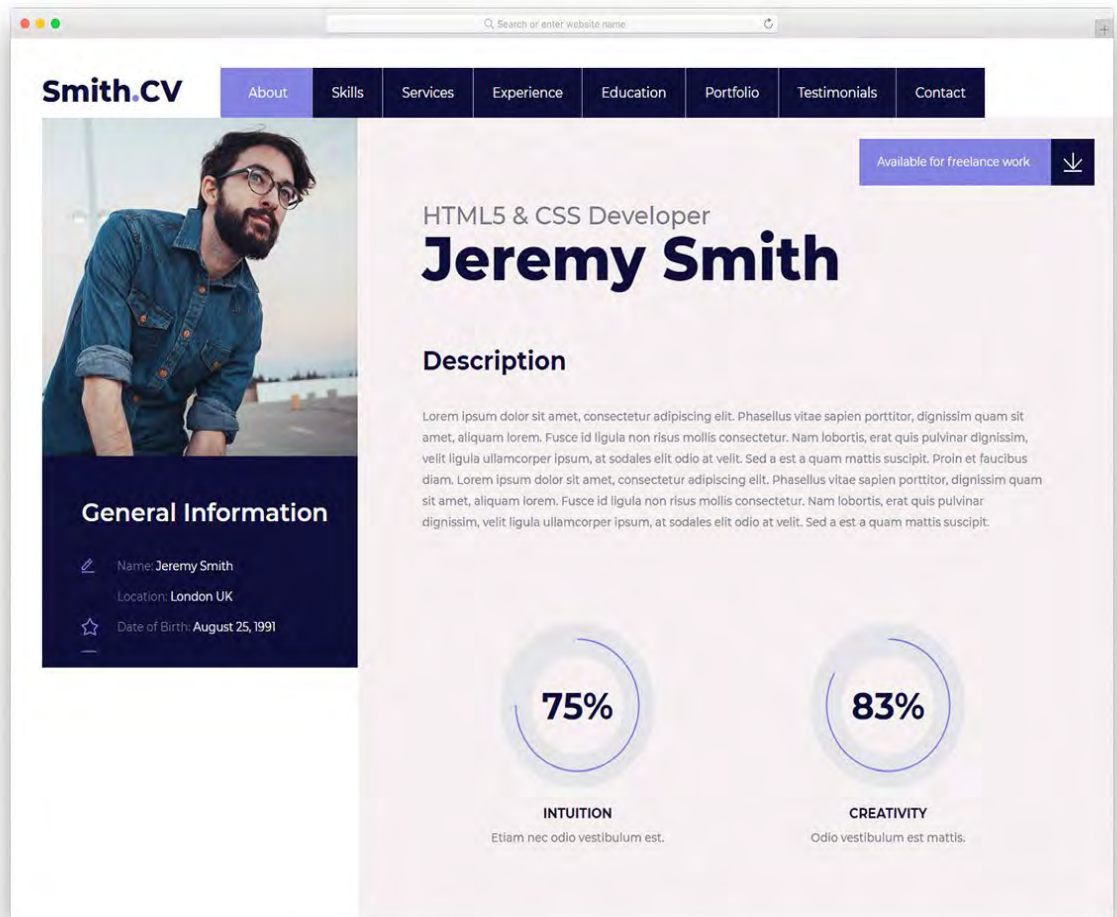
នៅពេលបច្ចុប្បន្ននេះគេឃើញមានវេបសាយយ៉ាងច្រើនសន្លឹកសន្លាប់ដែលបានបង្ហោះនៅ លើប្រព័ន្ធអ៊ីនធឺណិតដើម្បីបំពេញតម្រូវការ ឬចែករំលែកព័ត៌មានផ្សេងៗតាមប្រភេទដែលគេចង់បាន។ ខាងនេះគឺជាប្រភេទវេបសាយដែលមានលក្ខណៈផ្សេងៗគ្នា ។

១.៣.៤.១ Personal Website

Personal Website គឺជាប្រភេទគេហទំព័រមួយដែលត្រូវបានបង្កើតឡើងសម្រាប់បុគ្គលណាម្នាក់ ក្នុងគោលបំណងបង្ហាញព័ត៌មានផ្ទាល់ខ្លួនដើម្បីឲ្យអ្នកទស្សនាទាំងឡាយបានដឹង និងបានស្គាល់តាមរយៈវេបសាយដែលបានបង្កើតនេះ។ គេប្រើវាសម្រាប់បង្ហាញព័ត៌មានផ្ទាល់ខ្លួនដូចជា៖

- ប្រវត្តិរូបសង្ខេប
- បទពិសោធន៍ និងការងារ
- ចំណេះដឹង និងជំនាញ
- គោលបំណង

- សម្រាប់អ្នកប្រកបរបរ Freelancer ជាដើម ។



(រូបភាពទី ៥៖ Personal Website)

១.៣.៤.២ Information Website

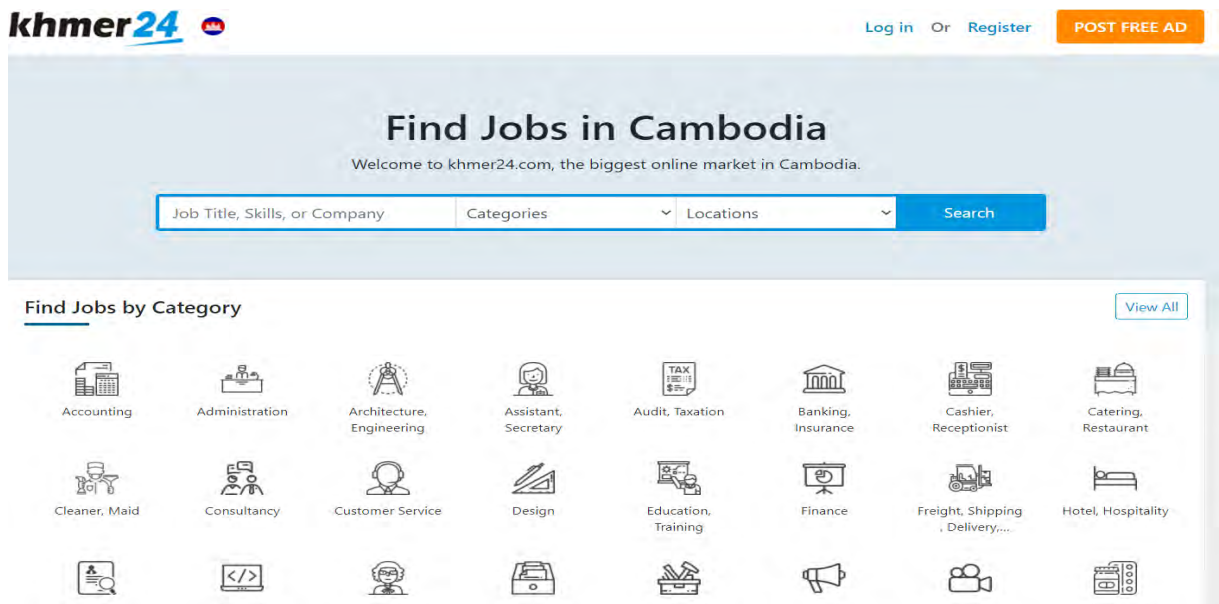
Information Website គឺជាប្រភេទគេហទំព័រមួយដែលត្រូវបានបង្កើតឡើងដោយបុគ្គល ក្រុម ហ៊ុន អង្គភាពរដ្ឋាភិបាល ឬស្ថាប័នណាមួយដែលដាក់បង្ហាញព័ត៌មានជាក់ស្តែងដូចជា ការងារ អប់រំ សមិទ្ធផល និងព័ត៌មានផ្សេងៗដែលទាក់ទង ។



(រូបភាពទី ៦ ៖ Information Website)

១.៣.៤.៣ Career Website

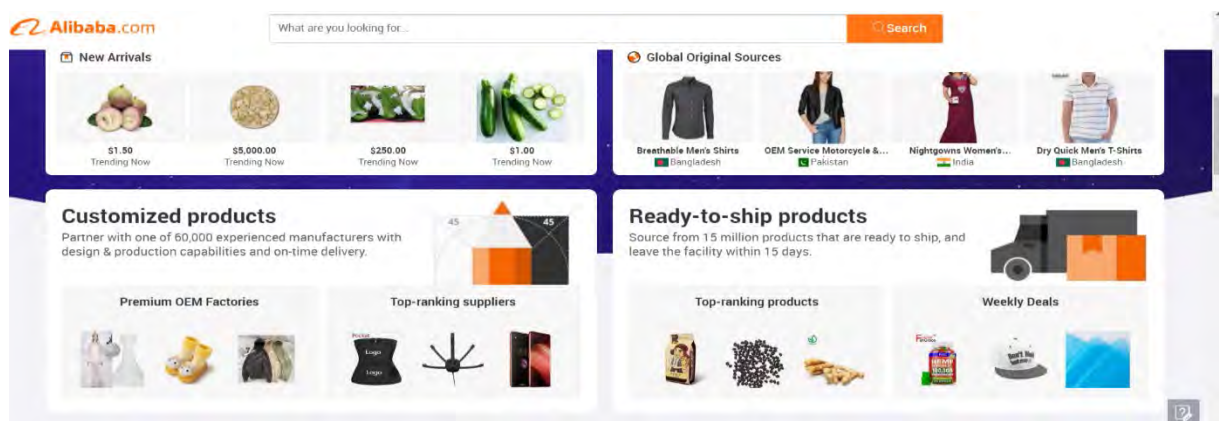
Career Website ជាប្រភេទគេហទំព័រមួយដែលត្រូវបានបង្កើតឡើងក្នុងគោលបំណងផ្សព្វផ្សាយជ្រើសរើសបុគ្គលិក នៅតាមក្រុមហ៊ុន ឬតាមស្ថាប័ននានា។ វេបសាយប្រភេទនេះជួយសម្រួលដល់អ្នកដែលកំពុងស្វែងរកការងារដោយមានភាពងាយស្រួល និង ចំណេញពេលវេលា។



(រូបភាពទី ៧៖ Career Website)

១.៣.៤.៤ E-Commerce Website

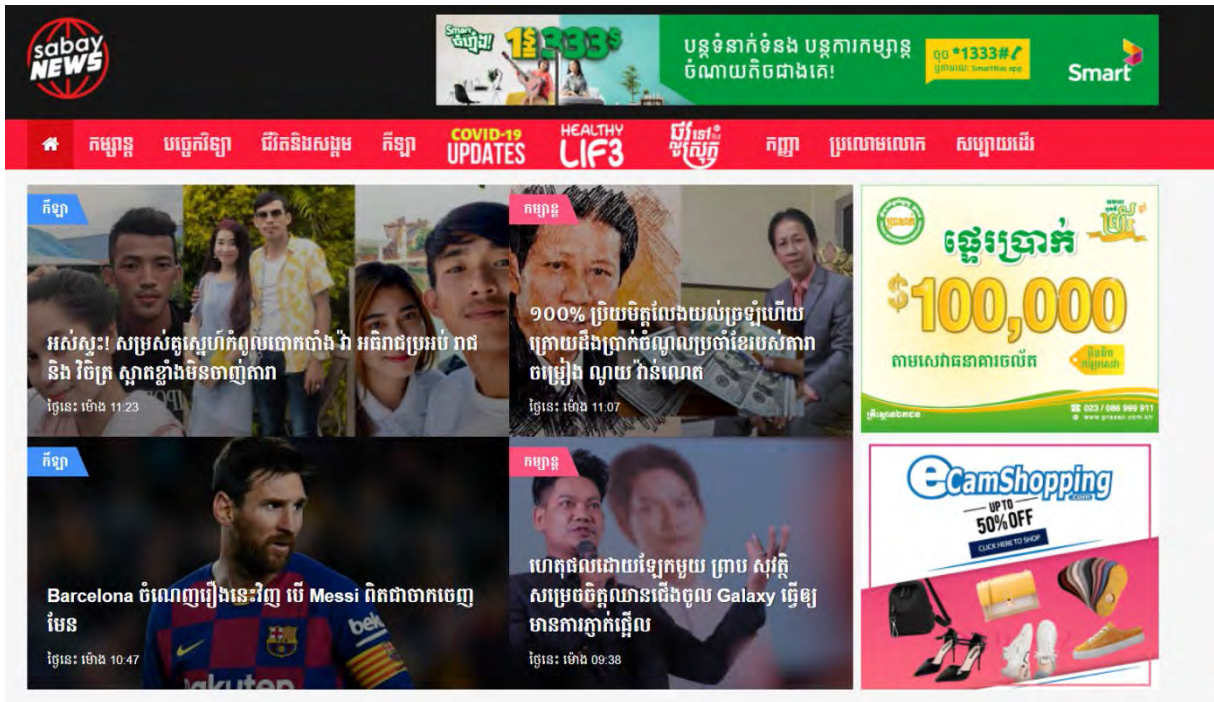
E-Commerce Website ប្រភេទគេហទំព័រមួយដែលត្រូវបានបង្កើតឡើងដោយក្រុមហ៊ុនអាជីវកម្មណាមួយ ក្នុងគោលបំណងបម្រើនូវសេវាកម្មអាជីវកម្មរបស់ខ្លួនតាមរយៈប្រព័ន្ធអ៊ីនធឺណិត។ នៅពេលដែលអ្នកទស្សនាចូលទៅកាន់គេហទំព័រប្រភេទនេះ គេអាចមើលឃើញព័ត៌មានកម្សៅផ្សេងៗនៅលើគេហទំព័រ ហើយគេអាចធ្វើការបញ្ជាទិញផលិតផលដែលដាក់បង្ហាញនៅលើគេហទំព័រនោះ។ បន្ទាប់ពីបានធ្វើការបញ្ជាទិញរួចរាល់ហើយក្រុមហ៊ុននឹងធ្វើការដឹកជញ្ជូនផលិតផលដែលបានបញ្ជាទិញនោះទៅកាន់អតិថិជន



(រូបភាពទី ៨៖ E-commerce Website)

១.៣.៤.៥ Entertainment & News Website

Entertainment & News Website គឺជាគេហទំព័រដែលគេប្រើសម្រាប់ចែករំលែកការកំសាន្ត និង ព័ត៌មានប្រចាំថ្ងៃ។



(រូបភាពទី ៩៖Entertainment & News Website)

១.៣.៥ ផ្ទៃទេវដា Home Page

Home Page គឺជា page ដំបូងនៃគេហទំព័រទាំងមូលដែលអ្នកទស្សនាមើលឃើញមុនគេពេលអ្នកទស្សនាចូលទៅកាន់គេហទំព័រ។ គេហទំព័រទាំងអស់សុទ្ធតែមាន Home Page ដែលអ្នកទស្សនាអាចចូលមើលតាមរយៈ Web Browser ។

១.៣.៦ ផ្ទៃទេវដា Domain Name

Domain Name គឺជាឈ្មោះសម្រាប់គេហទំព័រនីមួយៗ។ គេហទំព័រមួយអាចមាន Domain Name ផ្ទាល់បានលុះត្រាតែមានការចុះបញ្ជីសុំអនុញ្ញាតដើម្បីប្រើប្រាស់។ គេហទំព័រមួយមិនអាចមាន Domain Name ដូចគ្នាបានឡើយ។ Domain Name ទាំងអស់របស់គេហទំព័របន្ទាប់ពីបានចុះឈ្មោះរួចរាល់ហើយត្រូវបានបញ្ជូនទៅកាន់ Large Domain Name Register ដើម្បីរក្សាទុកនៅលើ DNS(Domain Name System Server)។

Domain Name ចែកចែកចេញជា ៣គឺ៖

- កម្រិត ១៖Top Level Domains(TLDS)Z មាន Domain ល្បីៗដូចជា .com .net .org ជាដើម។

- កម្រិត ២ និង កម្រិត៣៖ជាទូទៅទុកឲ្យអ្នកប្រើប្រាស់ចុងក្រោយបំពេញដោយខ្លួនឯងទៅតាមអ្វីដែលខ្លួនចង់បាន។

១.៣.៧ អ្វីទៅជា Web Browser

Web browser ជាកម្មវិធីមួយដែលមានលទ្ធភាព បកប្រែ ភាសា HTML ដែល គេសរសេរមកដើម្បីបង្កើតជាទំព័រផ្សេងៗ ដែលយើងបើកបានឃើញ រាល់ថ្ងៃតាមរយៈ Internet។ Web browser ដែលគេស្គាល់មានដូចជា៖

- Internet Explorer
- Mozilla Firefox
- Google Chrome
- Opera
- Safari

១.៣.៨ អ្វីទៅជា URL

URL មកពីពាក្យ Uniform Resource Locator គឺជាទីតាំងផ្ទាល់របស់ឯកសារនៅលើ Internet ។ដើម្បីចូលទៅកាន់ទីតាំងផ្ទាល់របស់ឯកសារនោះយើងត្រូវថាវាអាស័យដ្ឋានទាំងអស់ដែលយើងហៅថា URL ។

១.៣.៩ អ្វីទៅជា FTP

FTP មកពីពាក្យពេញថា File Transfer Protocol ដែលជួយឲ្យអ្នកប្រើប្រាស់អាចបញ្ជូន ឬផ្ទេរឯកសារ និងទិន្នន័យផ្សេងៗពី FTP Server យកមកទុកក្នុង Hard Disk របស់កុំព្យូទ័រតាមរយៈប្រព័ន្ធអ៊ីនធឺណិត។ FTP បានបែងចែកចេញជា ២ គឺ FTP Server និង FTP Client ។

- FTP Server គឺជាម៉ាស៊ីនកុំព្យូទ័រដែលផ្ទុកទិន្នន័យឬឯកសារផ្សេងៗសម្រាប់អ្នកប្រើប្រាស់អាចទាញយកមកបាន។

- FTP Clientជាម៉ាស៊ីនកុំព្យូទ័រដែលអាចបញ្ជូន Request ទៅកាន់ FTP Server ដើម្បីទាញយកទិន្នន័យឬឯកសារផ្សេងៗ។

១.៤ ព្រឹត្តិការណ៍ទូទៅនៃការអភិវឌ្ឍន៍

១.៤.១ អំពី HTML

HTML ដែលមានពាក្យពេញថា Hyper Text Markup Language ដែលគេប្រើប្រាស់វាដើម្បីបង្កើតវេបសាយ។ វាមិនមែនជា Programming Language តែវាជា Markup Language ។ Markup Language គឺជាសំណុំនៃ tags ជាច្រើនផ្គុំចូលគ្នាបង្កើតបានជា Web Page មួយ។ HTML មានផ្នែកសំខាន់ៗដូចជា HEAD, BODY, FOOTER ជាដើម។ ដើម្បីរក្សាទុក File ជាប្រភេទ HTML យើងត្រូវដាក់ extension .html ហើយគេអាចសរសេរវាដោយប្រើ IDE មួយចំនួនដូចជា Notepad, Adobe Dreamweaver, Studio Code, Subline Text និង IDE ជាច្រើនទៀត។

នេះគឺជាប្លង់ដំបូងរបស់ HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title> </title>
  </head>
  <body>
  </body>
</html>
```



(រូបភាពទី ១០៖ គំរូប្លង់ និង Logo សម្គាល់ HTML)

១.៤.២ អំពី CSS

CSS មានពាក្យពេញថា Cascading Style Sheet ដែលវាគឺជាភាសាមួយសម្រាប់ធ្វើការកំណត់លក្ខណៈ និងទ្រង់ទ្រាយរបស់វេបសាយ។ យើងអាចនិយាយយ៉ាងសាមញ្ញបានថាគឺជាភាសាសម្រាប់តុបតែង និងលម្អ HTML ដែលធ្វើឲ្យ interface មានលក្ខណៈស្រស់ស្អាត។ បន្ថែមលើនេះ CSS ធ្វើឱ្យមានភាពងាយស្រួលក្នុងការផ្លាស់ប្តូរ style នៅលើទំព័រជាច្រើនដោយពឹងផ្អែកលើ CSS តែមួយ។ CSS ចែកចេញជា ពីរប្រភេទគឺ Internal CSS និង External CSS។

- Internal CSS គឺជាការសរសេរកូដ CSS ក្នុង File ជាមួយគ្នានឹង HTML ដែលមានពីរយ៉ាងគឺ ការប្រើតាម style tag ជាការសរសេរ CSS statements នៅក្នុង tag style និងមួយទៀតគឺការប្រើតាម style attribute ក្នុង tag ជាការសរសេរ CSS statements នៅក្នុង tag ណាមួយរបស់ HTML។

- External CSS គឺជាការសរសេរកូដ CSS ក្នុង File។ វិធីដែលប្រើក្នុងការភ្ជាប់ទៅកាន់ CSS File មានពីរគឺ ការប្រើនូវ Link tag និងការប្រើ @import ក្នុង Style tag។

Inline CSS

```
<p style="color: blue;">This is a paragraph.</p>
```

Internal CSS

```
<head>
  <style type = text/css>
    body {background-color: blue;}
    p { color: yellow;}
  </style>
</head>
```

External CSS

```
<head>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
```

(រូបភាពទី ១១៖គំរូរូប្លង់ Internal និង External CSS)

១.៤.៣ អំពី Bootstrap

Bootstrap គឺជា Front-End Responsive Framework ដែលជាកូដមួយកញ្ចប់ដែលត្រូវបានគេចងក្រងឡើងដោយប្រើប្រាស់ ភាសា និងសមាសធាតុមួយចំនួនដូចជា Css, JavaScript, Font ជាដើម ហើយត្រូវបានគេយកមកប្រើប្រាស់ជាមួយ HTML ដើម្បីបង្កើតជា Template ដោយមិនចាំបាច់ប្រើប្រាស់ភាសា CSS ក៏បានដែរ។ Bootstrap អាចបង្កើត Website ដែលមានលក្ខណៈ Responsive (អាចបើកបាននៅក្នុង ទូរសព្ទ, អាយផេត និង កុំព្យូទ័រដោយមិនខូចទ្រង់ទ្រាយ)ហើយវាបានបង្កើត ជា Components (ធាតុ) ដែលចាំបាច់ប្រើជាច្រើន ហើយប្រសិនបើលោកអ្នកចង់ប្រើប្រាស់នូវ Components ណាមួយគឺគ្រាន់តែហៅឈ្មោះ Class របស់ Component

នោះមកដាក់ក្នុង Elements របស់ HTML រួចជាការស្រេច ដែល Components ទាំងនោះមាន ដូចជា Menu, Slider, Grid System, Category, Button, Form។



(រូបភាពទី ១២៖Bootstrap)

១.៤.៣.១ របៀបទាញយក Bootstrap មកប្រើប្រាស់

ដើម្បីយើងអាចប្រើប្រាស់ Bootstrap បានគឺមានពីរបៀបគឺ ទាញយក file Bootstrap ដោយផ្ទាល់មកដាក់ក្នុង Project រួចធ្វើការ link Script Bootstrap ចូលទៅកាន់ file html និង មួយទៀតគឺយើងអាច Copy CDN របស់ Bootstrap មកដាក់ក្នុង file html តែម្តង។ ដើម្បីទាញយក Bootstrap មកប្រើប្រាស់យើងអាចចូលទៅកាន់ វេបសាយរបស់ Bootstrap ឬក៏វេបសាយផ្សេងទៀត

<https://getbootstrap.com/> ។

- របៀបទី ១៖ Link Folder

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="bootstrap/css/bootstrap.min.css">
  <script src="libs/jquery/jquery.min.js"></script>
  <script src="libs/js/bootstrap.min.js"></script>
</head>
<body>

  <div class="container">

    </div>

  </body>
</html>
|
```

(រូបភាពទី ១៣៖ របៀបទី ១ នៃការប្រើប្រាស់ bootstrap)

- របៀបទី ២៖ CDN

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</head>
<body>

  <div class="container">
    <h1>My First Bootstrap Page</h1>
    <p>This is some text.</p>
  </div>

</body>
</html>
```

(រូបភាពទី ១៤៖ របៀបទី ២ នៃការប្រើប្រាស់ bootstrap)

១.៤.៣.២ សារប្រយោជន៍នៃ Bootstrap

បានជាយើងត្រូវតែ ចេះប្រើប្រាស់នូវ Framework Bootstrap នេះគឺដោយសារតែវាបាន បង្ក
នូវភាពងាយស្រួលដល់ Designer ដូចការបង្ហាញខាងក្រោម៖

- ចំណេញពេលវេលា
- កាត់បន្ថយការសរសេរ HTML និង CSS
- មានសមត្ថភាពខ្ពស់ជាង CSS
- មាន Component ច្រើនសម្រាប់ប្រើប្រាស់
- អាចបង្កើត Responsive Template បានយ៉ាងងាយ
- ងាយស្រួលក្នុងការ Design
- ងាយស្រួលក្នុងការបែងចែក Layout

១.៤.៤ អំពី Java Script

Java Script គឺជាភាសា Client Side Script មួយដែលផ្តល់ភាពងាយស្រួលក្នុងការបង្កើតវេប
សាយឲ្យមានលក្ខណៈរស់រវើក ដែលអាចធ្វើឲ្យប្រភព ឬ តួអក្សរផ្សេងៗមានចលនាបាន។ក្នុងការ
សរសេរកូដ Java Script គឺត្រូវបានចែកចេញជា ពីរប្រភេទគឺ Internal និង External។

ក.Internal Java Script

គឺជាការសរសេរកូដ Java Script នៅក្នុង file html ហើយកូដដែលត្រូវសរសេរគឺស្ថិតនៅក្នុង
tag script ដែលអាចស្ថិតនៅក្នុង HEAD ឬ BODY នៃ HTML Page។

ឧទាហរណ៍៖

```
<html>
  <head>
    <title>Internal JavaScript</title>
    <script type="text/javascript">
      document.write("This is Internal Javascript Example.!!!");
    </script>
  </head>
  <body>
  </body>
</html>
```

2. External Java Script

គឺជាការសរសេរកូដ Java Script នៅក្នុង File ផ្សេងគ្នាពី File HTML ដោយប្រើប្រាស់ extension .js ។ ដើម្បីប្រើប្រាស់ External Java Script ត្រូវ link file Java Script ចូល HTML File ។

ឧទាហរណ៍៖

file external.js

```
document.write("This is External Javascript Example.!!!");
```

home.html

```
<html>
  <head>
    <title>External JavaScript</title>
    <script type="text/javascript" src="external.js"></script>
  </head>
  <body>
  </body>
</html>
```

១.៤.៥ អំពី JQuery

JQuery គឺជា Library ដ៏មានឥទ្ធិពលមួយនៃភាសា Javascript ។ វាត្រូវបានបង្កើតឡើងដោយលោក John Resig និងបានបង្ហាញខ្លួនជាលើកដំបូងកាលពីឆ្នាំ 2006 ។ JQuery ត្រូវបានបង្កើតឡើងជាមួយនឹងគោលបំណងផ្តល់ភាពងាយស្រួលក្នុងប្រើប្រាស់ភាសា Javascript នៅលើគេហទំព័រ។ ជាមួយ JQuery អ្នកអាចសរសេរកូដត្រឹមតែមួយជួរដែលអាចធ្វើការងារស្មើនឹងកូដជាច្រើនជួរដែលបានសរសេរក្នុងភាសា Javascript ស្របនឹងពាក្យស្លោករបស់ JQuery «សរសេរតិច ធ្វើការច្រើន» ។

ក. របៀបតំឡើង JQuery

ការដំឡើងគឺមានលក្ខណៈងាយស្រួល ដោយសារ JQuery ក៏ជាកូដនៃភាសា Javascript ដែរ។ អ្នកអាចទាញយកវាមកទុក ឬប្រើប្រាស់តំណ CDN ដើម្បីបញ្ចូលវានៅក្នុងឯកសារ Html ។

- ចំពោះការទាញយក JQuery មករក្សាទុកនៅក្នុងកុំព្យូទ័រ <https://jquery.com/> ។ អ្នកអាចហៅវាមកប្រើប្រាស់នៅក្នុងឯកសារ Html ដូចខាងក្រោម៖

```
<head>
<script src="jquery-1.12.4.min.js"></script>
</head>
```

• សម្រាប់ការប្រើប្រាស់ jQuery តាមរយៈ CDN សូមបញ្ចូលតំណមួយក្នុងចំណោមតំណ
រទាំងពីរខាងក្រោមទៅក្នុងឯកសារ Html។

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/
jquery.min.js"></script>
</head>
```

Microsoft CDN៖

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/
jquery.min.js"></script>
</head>
```

Google CDN៖

```
<head>
<script src="https://ajax.aspnetcdn.com/ajax/jquery/
jquery-1.12.4.min.js"></script>
</head>
```

ខ. ការប្រើប្រាស់ JQuery

នេះគឺជាទម្រង់ទូទៅនៃកូដ jQuery: \$(selector).action() ។ វាចាប់ផ្តើមដោយការចាប់
យកធាតុនៃ Html ដោយប្រើ Selector រួចកំណត់សកម្មភាពដែលនឹងកើតឡើងលើធាតុនោះ។

ឧទាហរណ៍

```
$(document).ready(function(){
    $("p").click(function(){
        $(this).hide();
    });
});
```

គ. JQuery Selector

Selector គឺជាមុខងារដ៏សំខាន់មួយរបស់ jQuery ។ រាល់ Selector ទាំងអស់ចាប់ផ្តើមដោយសញ្ញាដុល្លារ និងរង់ក្រចកៈ \$() ។ វាត្រូវបានគេប្រើប្រាស់សម្រាប់ស្វែងរក និងជ្រើសរើសធាតុនៃ Html តាមរយៈ id name ឬ class ។

ឧទាហរណ៍

\$("#*"): ជ្រើសរើសគ្រប់ធាតុទាំងអស់។

\$(".nav"): ជ្រើសរើសធាតុដែលមាន class="nav" ។

\$("#p:first"): ជ្រើសរើសធាតុ <p> ដំបូងគេបង្អស់។

\$(":checkbox"): ជ្រើសរើសធាតុ Input ទាំងអស់ដែលមាន type="checkbox"។

\$("[href]"): ជ្រើសរើសធាតុទាំងអស់ដែលមាន href ជា attribute ។

\$("#p.article"): ជ្រើសរើសធាតុ <p> ទាំងអស់ដែលមាន class=article ។

គ. JQuery Event

Event ត្រូវបានគេប្រើដើម្បីឲ្យគេហទំព័រមួយមានលក្ខណៈបត់បែនទៅតាមការប្រើប្រាស់។ jQuery បានធ្វើឲ្យការប្រើប្រាស់ Event នៅលើគេហទំព័រកាន់តែមានភាពងាយស្រួលជាងមុនទៅទៀត។ ខាងក្រោមនេះគឺជាឧទាហរណ៍នៃការប្រើប្រាស់ Event ជាមួយ jQuery:

```
<script>
$("#p").click(function(){
    $(this).hide();
});
</script>
```

កូដខាងលើមានន័យថា នៅពេលដែលអ្នកចុចលើធាតុ <p> ណាមួយ នោះធាតុ <p> នោះនឹងត្រូវលាក់ទុកលែងបង្ហាញលើគេហទំព័រ។ មិនមែនមានតែ Event Click មួយប៉ុណ្ណោះទេ ខាងក្រោមនេះគឺជាឈ្មោះ Event ដែលត្រូវបានគេប្រើជាញឹកញាប់៖

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

(រូបភាពទី ១៥៖តារាង Event JQuery)

១.៤.៦ អំពី Laravel Framework

Laravel គឺជា open-source PHP framework ដែលបានបង្កើតឡើងដោយ **Taylor Otwell** ហើយមានបំណងអភិវឌ្ឍកម្មវិធីគេហទំព័រតាមគំរូ model-view-controller (MVC) architectural pattern និងផ្អែកលើ Symfony។

១.៤.៦.១ Server Requirements សម្រាប់ laravel 6.0

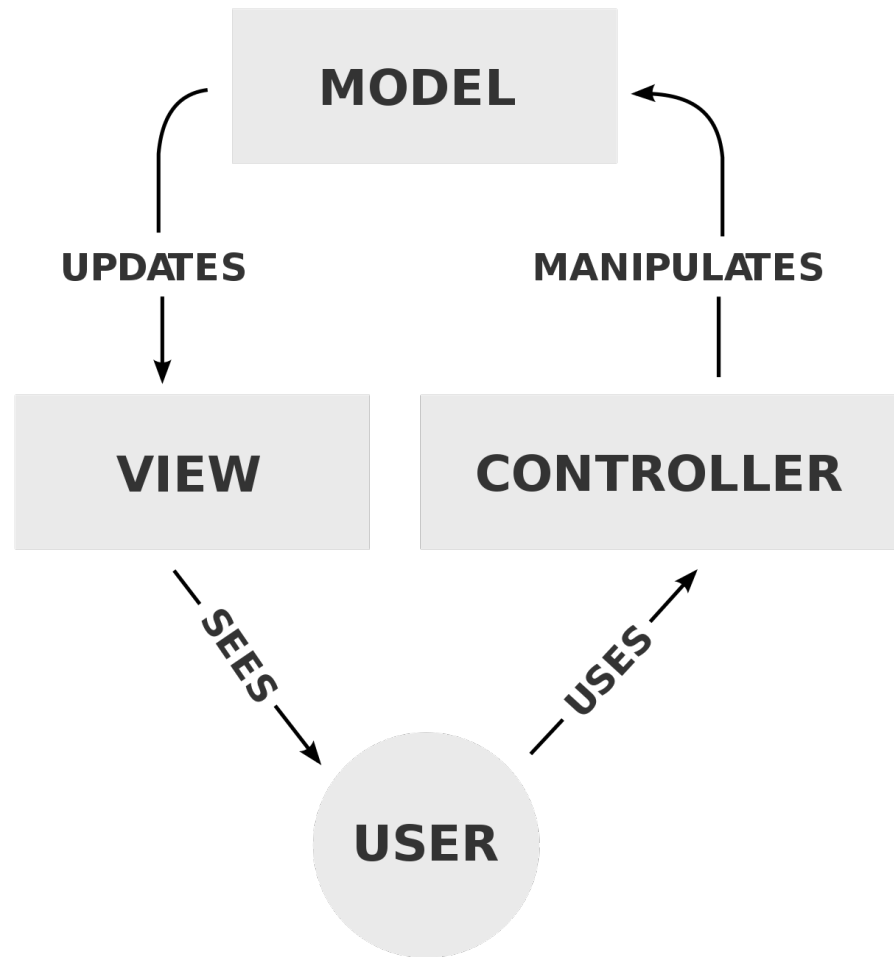
- PHP >= 7.2.0
- BCMath PHP Extension
- Ctype PHP Extension
- Fileinfo PHP extension
- JSON PHP Extension
- Mbstring PHP Extension
- OpenSSL PHP Extension
- PDO PHP Extension
- Tokenizer PHP Extension
- XML PHP Extension

១.៤.៦.២ Model-View-Controller (MVC)

Model-View-Controller គឺជាលំនាំស្ថាបត្យកម្មដែលត្រូវបានប្រើជាទូទៅសម្រាប់ការអភិវឌ្ឍន៍ user interfaces ដែលបែងចែកកម្មវិធីជាបីផ្នែកដែលទាក់ទងគ្នា។

- Controller: គឺដើម្បីដោះស្រាយ user requests ហើយនិងទៅយកទិន្នន័យដោយប្រើប្រាស់ Models

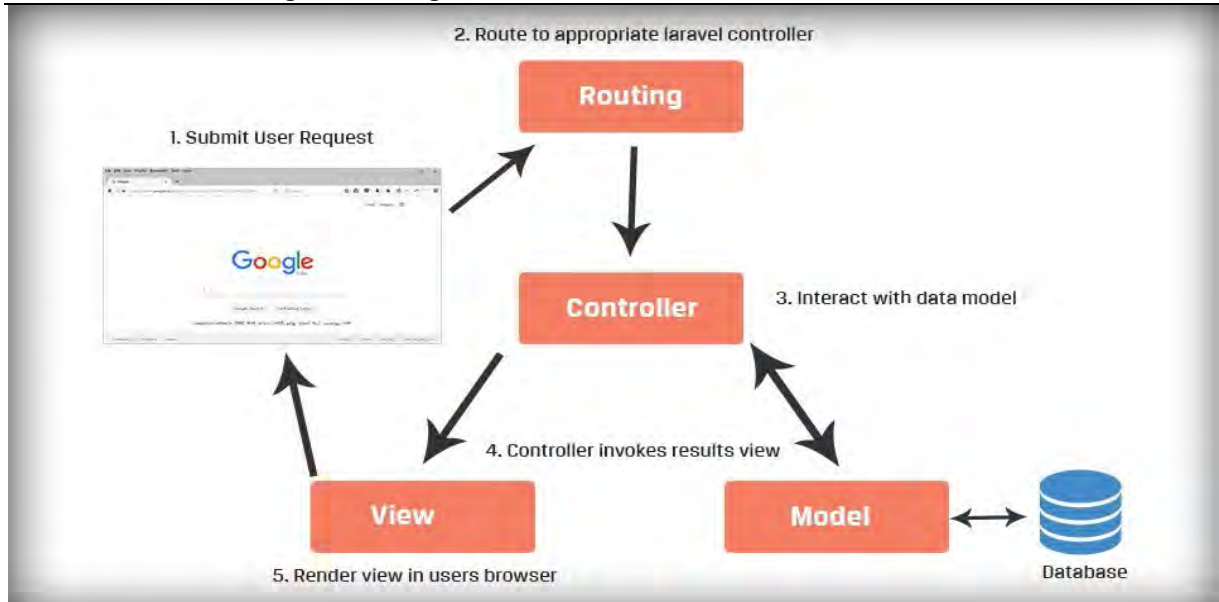
- Model: គឺដើម្បីធ្វើការទាក់ទងគ្នាជាមួយ database ហើយនិងទាយយកទិន្នន័យ
- View: គឺដើម្បីបង្ហាញគេហទំព័រ។



(រូបភាពទី ១៦៖ទម្រង់ MVC របស់ Laravel)

១.៤.៦.៣ របៀប Laravel ធ្វើការ

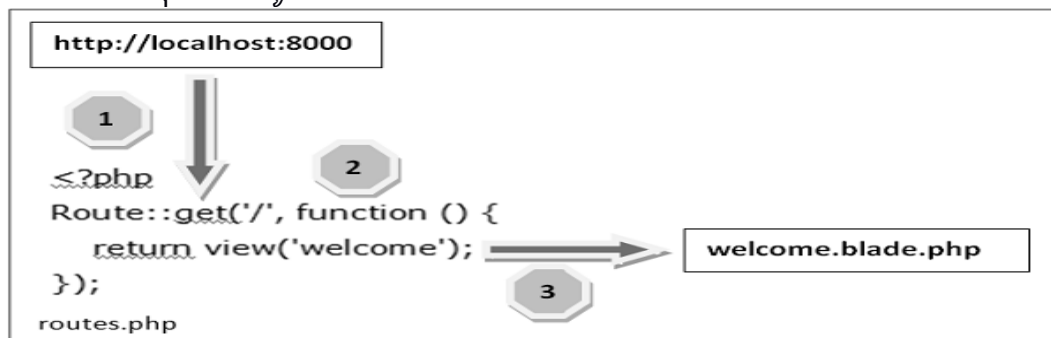
របៀបដែល Laravel ធ្វើការជាដំបូង user ធ្វើការ request ពី browser តាមរយៈ url ទៅកាន់ route បន្ទាប់ធ្វើការ access បន្តទៅកាន់ controller ហើយ controller ធ្វើការបញ្ជូនទៅកាន់ model ដែល model ជាអ្នកប្រស្រ័យទាក់ទងជាមួយ database រួច database បញ្ជូនទិន្នន័យមកអោយ model ហើយ model ចាប់ផ្តើមបញ្ជូនបន្តមកកាន់ controller បន្ទាប់មក controller ហៅ view ដែលធ្វើការ return ទិន្នន័យតាមរយៈ view រួចមក view ជាអ្នកបង្ហាញលទ្ធផលនៅក្នុង browser របស់អ្នកប្រើប្រាស់។ ដែលបានបង្ហាញដូចរូបខាងក្រោម៖



(រូបភាពទី ១៧:របៀប Laravel ធ្វើការ)

១.៤.៦.៤ Routing

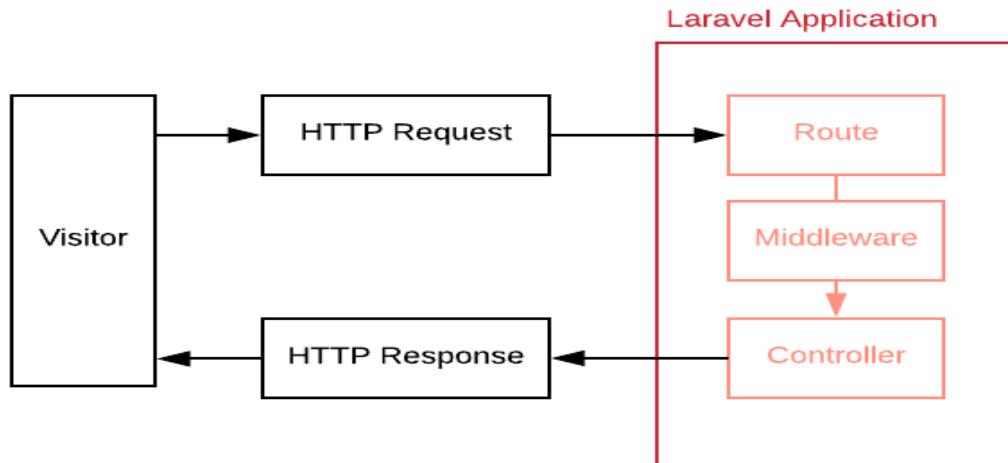
នៅក្នុង laravel គ្រប់ការ request ទាំងអស់ គឺត្រូវបានធ្វើការជាមួយ route ទាំងអស់ដែល route ជាអ្នកជួយក្នុងការបញ្ជូន request ទៅកាន់ controller។



(រូបភាពទី ១៨:បង្ហាញពីទម្រង់របស់ Router)

១.៤.៦.៥ Middleware

Middleware គឺជាស្ថានភាពរវាង request និង response។ Middleware អាចធ្វើការត្រួតពិនិត្យលក្ខខណ្ឌណាមួយរបស់អ្នកប្រើប្រាស់ ដើម្បីអនុញ្ញាតអោយមានសំណើរបស់ទៅក្នុងកម្មវិធី។



(រូបភាពទី ១៩៖ដំណើរការរបស់ Middleware)

Command បង្កើត middleware:

-> php artisan make:middleware CheckAge

```

public function handle($request, Closure $next)
{
    if ($request->user()->age <= 18) {
        return redirect('home');
    }

    return $next($request);
}
    
```

(រូបភាពទី ២០៖ទម្រង់ Middleware)

ដើម្បីអោយ middleware ដំណើរការជាពេលដែល HTTP request ទៅកាន់កម្មវិធី យើងត្រូវ assign middleware key មួយទៅក្នុង app/Http/Kernel.php

```
// Within App\Http\Kernel Class...
```

```
protected $routeMiddleware = [
    'auth' => \App\Http\Middleware\Authenticate::class,
    'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
    'bindings' => \Illuminate\Routing\Middleware\SubstituteBindings::class,
    'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
    'can' => \Illuminate\Auth\Middleware\Authorize::class,
    'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
    'signed' => \Illuminate\Routing\Middleware\ValidateSignature::class,
    'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
    'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
];
```

(រូបភាពទី ២១៖បង្ហាញពី របៀប assign middleware key ទៅក្នុង Kernel.php)

១.៤.៦.៦ HTTP Request

ដើម្បីទទួលបាន HTTP request តាមរយៈ dependency injection អ្នកគួរតែវាយពាក្យជំនួយ Illuminate\Http\Request class ទៅក្នុង controller method ។ request instance ត្រូវបានដាក់បញ្ចូលដោយស្វ័យប្រវត្តិដោយ service container ។

```
<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Http\Requests;
use App\Http\Controllers\Controller;

class UserRegistration extends Controller {
    public function postRegister(Request $request) {
        //Retrieve the name input field
        $name = $request->input('name');
        echo 'Name: ' . $name;
        echo '<br>';

        //Retrieve the username input field
        $username = $request->username;
        echo 'Username: ' . $username;
        echo '<br>';

        //Retrieve the password input field
        $password = $request->password;
        echo 'Password: ' . $password;
    }
}
```

(រូបភាពទី ២២៖បង្ហាញពី ការប្រើ HTTP request)

១.៤.៦.៧ HTTP Response

គ្រប់ route និង controller ទាំងអស់គឺសុទ្ធតែធ្វើការ response ត្រឡប់ទៅកាន់ browser របស់អ្នកប្រើប្រាស់។ laravel បានផ្តល់នូវវិធីផ្សេងៗគ្នាជាច្រើនដើម្បី response។ ការបង្កើតការ response មាន response ជា string, array, json ហើយនិងការ redirect ទៅកាន់ផ្នែកណាមួយជាដើម។

១.៤.៦.៨ Validator

Laravel ផ្តល់នូវវិធីសាស្ត្រផ្សេងៗគ្នាជាច្រើនដើម្បីធ្វើឱ្យទិន្នន័យចូលកម្មវិធីរបស់អ្នកមាន validate ត្រឹមត្រូវ។ វាទាមទារឱ្យអ្នកប្រើប្រាស់បំពេញបែបបទឱ្យបានត្រឹមត្រូវនៅក្នុង Form ។

- ប្រើប្រាស់ `$request validate`

```
public function contactStore(Request $request) {
    // Validate and store the contact...
    $request->validate([
        'name' => 'required|min:2',
        'email' => 'required|email',
        'phone' => 'required'
    ]);
}
```

(រូបភាពទី ២៣: បង្ហាញពី ការ validate តាម `$request validate`)

- ប្រើប្រាស់ Validator instance

```
Validator::make($request->all(), [
    'title' => 'required|unique:posts|max:255',
    'body' => 'required',
])->validate();
```

(រូបភាពទី ២៤: បង្ហាញពី ការ validate តាម Validator instance)

១.៤.៦.៩ View

Views គឺកន្លែងផ្ទុក HTML ហើយវាអភិរក្សទុកនៅក្នុង resources/views directory ។ view ដែលសាមញ្ញគឺដូចខាងក្រោម៖

```
<!-- View stored in resources/views/greeting.blade.php -->

<html>
  <body>
    <h1>Hello, {{ $name }}</h1>
  </body>
</html>
```

```
Route::get('/', function () {
    return view('greeting', ['name' => 'James']);
});
```

(រូបភាពទី ២៥៖បង្ហាញពី ទម្រង់របស់ View)

១.៤.៦.១០ Database configuration

ដើម្បីឲ្យ Database អាច connect បានជាមួយ Project យើងត្រូវចូលទៅកាន់ file .env នៅក្នុង project រួចធ្វើការបំពេញដូចជា Database Name, Host, Username, Password ឲ្យបានត្រឹមត្រូវ។

```
#Supports: MySQL, SQL Server, SQLite, and PostgreSQL.
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=your_database_name
DB_USERNAME=your_db_username
DB_PASSWORD=your_db_password
```

(រូបភាពទី ២៦៖បង្ហាញពី ការ connection database ក្នុង .env file)

១.៤.៦.១១ Database Migration

Migrations ត្រូវបានប្រើដើម្បីរក្សាទុកព័ត៌មានលំអិតអំពី database table ហើយវាមានលក្ខណៈសម្បត្តិដូច្នេះអ្នកមិនចាំបាច់បង្កើត table ទាំងអស់ដោយ manually ទេដោយចូលទៅកាន់ database interface ឬមួយដូចជា phpmyadmin ឡើយ។ យើងអាចបង្កើត migrations ដោយប្រើ artisan ជាមួយ command "make: migration"៖

```
php artisan make:migration create_todos_table
```

```

1  <?php
2
3  use Illuminate\Support\Facades\Schema;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Database\Migrations\Migration;
6
7  class CreateTodosTable extends Migration
8  {
9      public function up()
10     {
11         Schema::create('todos', function (Blueprint $table) {
12             $table->bigIncrements('id');
13             $table->timestamps();
14         });
15     }
16
17     public function down()
18     {
19         Schema::dropIfExists('todos');
20     }
21 }
```

(រូបភាពទី ២៧៖បង្ហាញពីទម្រង់របស់ Migration)

មុននឹង running migration command យើងត្រូវបន្ថែម line របស់ code នៅក្នុង AppServiceProvider.php នៅក្នុង boot() function:

```
\Schema::defaultStringLength(191);
```

រួចហើយត្រូវ run migration command

```
php artisan migrate
```

១.៤.៦.១២ Database Query Builder

Database query builder របស់ laravel វាផ្តល់ភាពងាយស្រួលក្នុងការ running និង creating database queries។ វាអាចប្រើដើម្បីធ្វើ database operations នៅក្នុងកម្មវិធីរបស់យើងហើយវាធ្វើការ support គ្រប់ database systems។ ដើម្បីប្រើប្រាស់ database query builder យើងត្រូវ import: Illuminate\Support\Facades\DB។

- Insert

```
DB::table('users')->insert([
    ['email' => 'john@example.com', 'votes' => 0]
]);
```

- Select

```
$users = DB::table('users') ;
```

```
$users = DB::table('users')->select('name', 'email as user_email')->get();
```

- Update

```
DB::table('users')
    ->where('id', 1)
    ->update(['votes' => 1]);
```

- Delete

```
DB::table('users')->delete();
```

```
DB::table('users')->where('votes', '>', 100)->delete();
```

១.៤.៦.១៣ Eloquent ORM

រាល់ database table នីមួយៗគឺមាន "Model" ដែលបានប្រើប្រាស់ដើម្បីធ្វើការទំនាក់ទំនងជាមួយ table នោះ។ Model អនុញ្ញាតអោយអ្នកធ្វើការ query ទិន្នន័យនៅក្នុង table ដូចជាការបញ្ចូល record ថ្មីទៅក្នុង table។

- Create Model

```
php artisan make:model Flight
```

```
namespace App;

use Illuminate\Database\Eloquent\Model;

class Flight extends Model
{
    /**
     * The table associated with the model.
     *
     * @var string
     */
    protected $table = 'my_flights';
}
```

- Create Model with Migration

```
php artisan make:model Flight --migration
```

```
php artisan make:model Flight -m
```

- Insert

```
$flight = new Flight;

$flight->name = $request->name;

$flight->save();
```

- Select

```
$flight = App\Flight::where('number', 'FR 900')->first();
```

```
$flights = App\Flight::all();

foreach ($flights as $flight) {
    echo $flight->name;
}
```

- Update

```
$flight = App\Flight::find(1);

$flight->name = 'New Flight Name';

$flight->save();
```

- Delete

```
$flight = App\Flight::find(1);

$flight->delete();
```

១.៤.៧ មេរៀនសង្ខេប PHP**១.៤.៧.១ និយមន័យ**

PHP ត្រូវបានផ្តល់ឈ្មោះជាផ្លូវការ Hyper Text Preprocessor ដែលមានលក្ខណៈជា Object-Oriented Language ដែលដំណើរការនៅលើ server ។ ជាទូទៅគេសរសេរក្នុងបរិបទ HTML នៅក្នុង PHP Script ហើយវាមិនត្រូវបានបញ្ជូនទៅឲ្យ Client ដោយ server ផ្ទាល់នោះទេវាធ្វើការបញ្ជូនដោយ PHP Engine ។

១.៤.៧.២ អថេរ (Variable)

Variable ជាគ្រឹះដ៏សំខាន់សម្រាប់ភាសាកុំព្យូទ័រ វាត្រូវបានគេប្រើប្រាស់ដើម្បីផ្ទុកនូវតម្លៃបណ្តោះអាសន្ននៅពេលដែលប្រតិបត្តិការ Script ម្តងៗ ។ សម្រាប់ PHP ឈ្មោះរបស់ Variable ត្រូវបានចាប់ផ្តើមដោយនិមិត្តសញ្ញា \$ (Dollar Sign) អាចផ្តល់ឈ្មោះរបស់ variable ជាតួអក្សរ ឬ underscore (_) ។ យើងមិនអាចប្រើប្រាស់អក្សរដកឃ្លាជាឈ្មោះរបស់ variable បានទេ ។

ឧទាហរណ៍៖ការផ្តល់ឈ្មោះ variable ដែលត្រឹមត្រូវ

\$sok ;

\$sok_som_neang ;

\$_Dara ;

\$Chenda22 ;

ឧទាហរណ៍៖ការផ្តល់ឈ្មោះ variable ដែលមិនត្រឹមត្រូវ

\$123 ;

\$*ABC ;

\$A+B ;

បញ្ជាក់៖រាល់ចុងបញ្ចប់នៃរបាយការណ៍ PHP នីមួយៗត្រូវតែបញ្ចប់ដោយ (;) semicolon ។

១.៤.៧.៣ Data Type

ប្រភេទខុសគ្នានៃទិន្នន័យត្រូវបានប្រើប្រាស់នូវចំនួនសរុបរបស់អង្គចងចាំផ្សេងៗគ្នា ហើយវាអាចប្រព្រឹត្តទៅនៅពេលដែលអ្នករៀបចំវាក្នុង script របស់អ្នក។ មានភាសាកម្មវិធីមួយចំនួនទាមទារឲ្យអ្នកសរសេរកម្មវិធី ប្រកាសនូវប្រភេទទិន្នន័យពីមុខ variable ដោយឡែកភាសា PHP ការប្រើប្រាស់ variable គឺមានភាពងាយស្រួល ដោយវានឹងធ្វើការគណនានូវប្រភេទទិន្នន័យដោយស្វ័យប្រវត្តិនៅពេលដែលអ្នកបានផ្តល់តម្លៃឲ្យវា ។

Standard Data Type		
ប្រភេទ	តម្លៃ	ពិពណ៌នា
Integer	5	ផ្ទុកនូវតម្លៃលេខជាចំនួនគត់
Double	5.643	ផ្ទុកនូវតម្លៃលេខជាចំនួនទសភាគ
String	"Hello world"	ផ្ទុកនូវតម្លៃជាបណ្តុំនៃតួអក្សរ
Boolean	true	ផ្ទុកនូវតម្លៃពិសេស ពិត ឬ មិនពិត
Array		ផ្ទុកនូវតម្លៃមានលក្ខណៈជា Array
Object		ផ្ទុកនូវតម្លៃមានលក្ខណៈជា Object

(រូបភាពទី២៨ ៖ តារាងបញ្ជាក់ Data Type របស់ PHP)

១.៤.៧.៤ Operator

ក. Arithmetic Operators

Arithmetic Operators ត្រូវបានប្រើដើម្បីអនុវត្តប្រតិបត្តិការនេឡិចទ្រីទៅដូចជា ការបូកលេខ ដកលេខ គុណលេខ និងចែកលេខជាដើម។

Operator	Name	Example	Explanation
+	Addition	\$a + \$b	Sum of operands
-	Subtraction	\$a - \$b	Difference of operands
*	Multiplication	\$a * \$b	Product of operands
/	Division	\$a / \$b	Quotient of operands
%	Modulus	\$a % \$b	Remainder of operands

(រូបភាពទី ២៩៖ តារាងពិពណ៌នា Arithmetic Operator)

ខ. Assignment Operators

Assignment Operators ប្រើដើម្បីផ្តល់នូវតម្លៃទៅឲ្យ variable ។ Assignment Operator ត្រូវបាន initialize variable គ្រប់ពេលវាគឺប្រើប្រាស់សញ្ញា (= ។ Assignment Operator ចាប់យកតម្លៃពី operand ខាងស្តាំទៅឲ្យ operand ខាងឆ្វេង។

Operator	Name	Example	Explanation
=	Assign	\$a = \$b	The value of right operand is assigned to the left operand.
+=	Add then Assign	\$a += \$b	Addition same as \$a = \$a + \$b
-=	Subtract then Assign	\$a -= \$b	Subtraction same as \$a = \$a - \$b
*=	Multiply then Assign	\$a *= \$b	Multiplication same as \$a = \$a * \$b
/=	Divide then Assign (quotient)	\$a /= \$b	Find quotient same as \$a = \$a / \$b
%=	Divide then Assign (remainder)	\$a %= \$b	Find remainder same as \$a = \$a % \$b

(រូបភាពទី ៣០៖ តារាងពិពណ៌នា Assignment Operator)

គ. Comparison Operators

Comparison Operator ប្រើដើម្បីប្រៀបធៀបនូវ operand ទាំងឡាយ ដោយផ្តល់នៅតម្លៃត្រឡប់ជា Boolean(True or False) ។

Operator	Name	Example	Explanation
==	Equal	\$a == \$b	Return TRUE if \$a is equal to \$b
===	Identical	\$a === \$b	Return TRUE if \$a is equal to \$b, and they are of same data type
!==	Not identical	\$a !== \$b	Return TRUE if \$a is not equal to \$b, and they are not of same data type
!=	Not equal	\$a != \$b	Return TRUE if \$a is not equal to \$b
<>	Not equal	\$a <> \$b	Return TRUE if \$a is not equal to \$b
<	Less than	\$a < \$b	Return TRUE if \$a is less than \$b
>	Greater than	\$a > \$b	Return TRUE if \$a is greater than \$b
<=	Less than or equal to	\$a <= \$b	Return TRUE if \$a is less than or equal \$b
>=	Greater than or equal to	\$a >= \$b	Return TRUE if \$a is greater than or equal \$b
<=>	Spaceship	\$a <=> \$b	Return -1 if \$a is less than \$b Return 0 if \$a is equal \$b Return 1 if \$a is greater than \$b

(រូបភាពទី ៣១៖ តារាងពិពណ៌នា Comparison Operator)

២៧. Logical Operators

Logical Operator បំប្លែង operand ឲ្យទៅជាតម្លៃ Boolean រួចធ្វើការប្រៀបធៀបតម្លៃទាំងនោះ។

Operator	Name	Example	Explanation
and	And	\$a and \$b	Return TRUE if both \$a and \$b are true
Or	Or	\$a or \$b	Return TRUE if either \$a or \$b is true
xor	Xor	\$a xor \$b	Return TRUE if either \$ or \$b is true but not both
!	Not	! \$a	Return TRUE if \$a is not true
&&	And	\$a && \$b	Return TRUE if either \$a and \$b are true
	Or	\$a \$b	Return TRUE if either \$a or \$b is true

រូបភាពទី ៣២៖ តារាងពិពណ៌នា Comparison Operator

២៨. Incrementing and Decrementing Operators

Incrementing and Decrementing Operators ប្រើដើម្បីបង្កើន ឬ បន្ថយតម្លៃរបស់ variable ដែលជា Integer ហើយជាទូទៅប្រើដើម្បីរាប់ iteration របស់ loop ។

Operator	Name	Example	Explanation
++	Increment	++\$a	Increment the value of \$a by one, then return \$a
		\$a++	Return \$a, then increment the value of \$a by one
--	decrement	--\$a	Decrement the value of \$a by one, then return \$a
		\$a--	Return \$a, then decrement the value of \$a by one

(រូបភាពទី ៣៣៖ តារាងពិពណ៌នា Incrementing and Decrementing Operator

១.៤.៧.៥ Control Statement

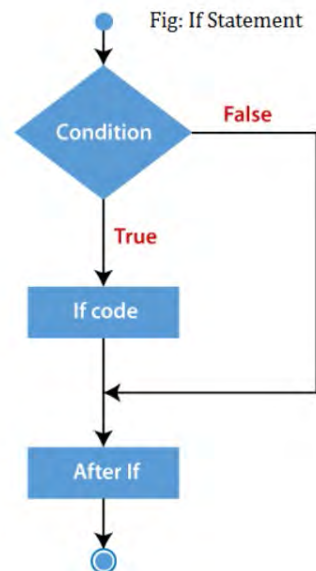
ក. If Statement

If Statement គឺជាវិធីសាស្ត្រដែលប្រើដើម្បីត្រួតពិនិត្យទៅលើប្រតិបត្តិការរបស់ Statement ដែលនៅបន្ទាប់វា (អាចជា Single statement ដែលនៅក្នុងសញ្ញា {statement} ។ If Statement ធ្វើការ វាយតម្លៃ Condition ប្រសិនបើ Condition របស់ If ផ្តល់តម្លៃ true នោះ statement ដែលនៅខាង ក្រោមនឹងត្រូវបានអនុវត្ត។

Syntax របស់ If Statement

```
If(condition)
{
    កូដដែលត្រូវដំណើរការ
}
```

(រូបភាពទី ៣៥៖ Flowchart If statement)



ខ. If-else-if Statement

អ្នកអាចប្រើប្រាស់នូវទម្រង់ if-else-if statement ដើម្បីធ្វើការពិនិត្យលើ Expression មុន ពេលដែល Script ត្រូវបានអនុវត្តនូវ default block of code។

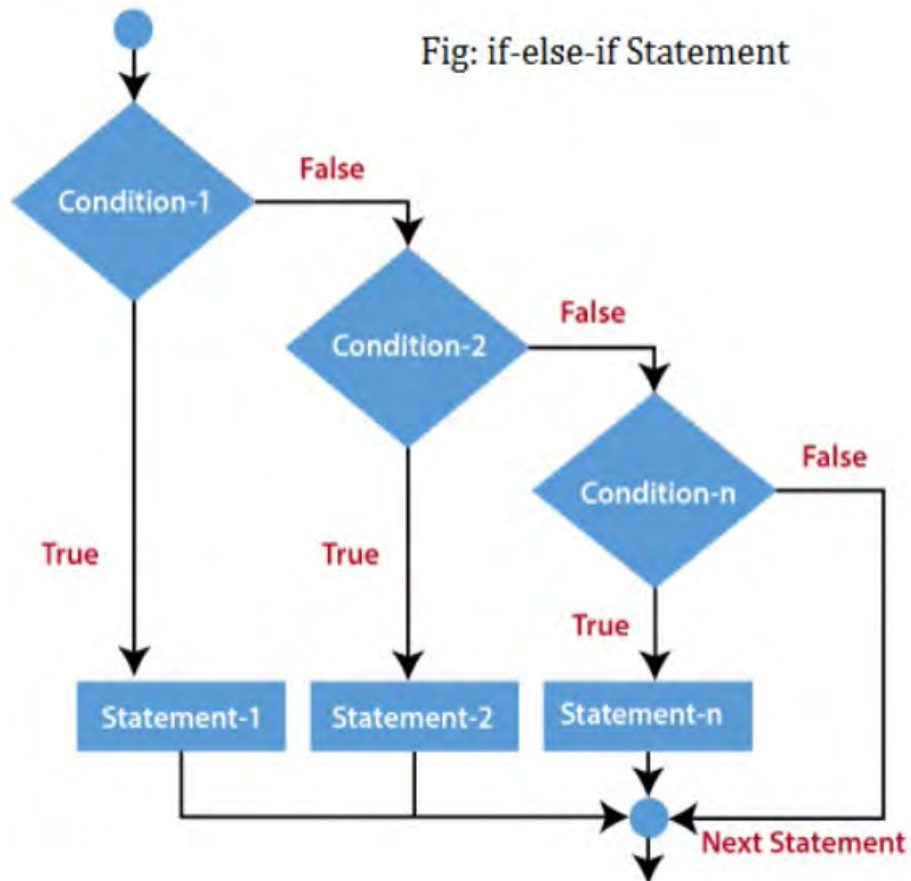
Syntax របស់ If-else-if statement

```
If(condition)
{
    //code ដែលត្រូវអនុវត្តន៍ បើសិនជា condition ផ្តល់តម្លៃពិត
}
else if (another condition )
{
    //code ដែលត្រូវអនុវត្តន៍ បើសិនជា condition ដំបូងផ្តល់តម្លៃមិនពិត ហើយ condition វា
    ផ្តល់តម្លៃពិត
```

}

else{//code ដែលត្រូវអនុវត្តន៍ បើសិនជាគ្មាន condition ណាមួយពិត

}



(រូបភាពទី ៣៥៖ Flowchart If-else-if statement)

គ. Loop

Loop Statement អាចឲ្យអ្នកអនុវត្តន៍ការងារម្តងហើយម្តងទៀត ក្នុង project របស់អ្នករហូតដល់វាសម្រេចលក្ខណ្ឌ ឬអ្នកបញ្ជាឲ្យចាកចេញពី loop ។

- For Loop

```
for(initialization; condition; increment/decrement){
    //code to be executed
}
```

រាល់ expression នីមួយៗដែលនៅក្នុងរង្វង់ក្រចក របស់ for statement គឺត្រូវបែកចែកដោយ semicolon(;)^១ expression ទី១ ចាប់ផ្តើមរាប់ variable ហើយ expression ទី២ធ្វើការត្រួតពិនិត្យលក្ខណ្ឌ និង expression ទី៣ បង្កើន ឬបន្ថយចំនួនដែលរាប់។

ឧទាហរណ៍៖

```
<!DOCTYPE>
<html>
<head>
<title>Web page title</title>
</head>
<body>
    <?php
        for($n=1;$n<=10;$n++){
            echo "$n<br/>";
        }
    ?>

</body>
</html>
```

- While Loop

While loop គឺជាប្រភេទមួយនៃ loop។ expression របស់វាផ្តល់ជាតម្លៃ true ឬ false ដូចច្នេះប្រសិន expression ផ្តល់តម្លៃ true នោះកូដនឹងត្រូវអនុវត្តន៍ ហើយកូដដែលស្ថិតនៅក្នុង loop នោះត្រូវបានគេឲ្យឈ្មោះថា iteration។

ឧទាហរណ៍៖

```

<!DOCTYPE>

<html>
<head>
<title>Web page title</title>
</head>
<body>
    <?php
        $n=1;
        while($n<=10){
            echo "$n<br/>";
            $n++;
        }
    ?>

</body>
</html>

```

- Do...While Loop

វាមានលក្ខណៈប្រហាក់ប្រហែលទៅនឹង while loop ដែរ ប៉ុន្តែលក្ខណៈខុសពីគ្នាគឺ កូដរបស់ do while គឺអនុវត្តន៍មុនពេល លក្ខខណ្ឌត្រូវបានត្រួតពិនិត្យថា ។

ឧទាហរណ៍៖

```

<!DOCTYPE>

<html>
<head>
<title>Web page title</title>
</head>
<body>
    <?php
        $n=1;
        do{
            echo "$n<br/>";
            $n++;
        }while($n<=10);
    ?>

</body>
</html>

```

១.៤.៧.៦ Function

Function គឺជាបន្តនៃ block code ដែលអ្នកបង្កើត ហើយអនុញ្ញាតអោយអ្នកហៅ យកមកប្រើ នៅក្នុង script របស់អ្នក។ នៅពេលដែលយើងហៅ function ក្នុង ដែលនៅខាងក្នុងនឹងត្រូវអនុវត្តន៍ ហើយយើងក៏អាច បញ្ជូនតម្លៃទៅអោយ function ឬទទួលតម្លៃពី function មកវិញ។

ក. Calling Functions

Function មានពីរប្រភេទគឺ function ដែលមានស្រាប់ ភ្ជាប់ជាមួយនឹងភាសាភូមិវិធី និង function ដែលយើងបង្កើតឡើងដោយខ្លួនឯង។

ខ. Defining Functions

យើងអាចបង្កើត function ដោយប្រើ function statement។

ឧទាហរណ៍៖

```
<?php
    function sayHello($anme){
        echo "សួស្តី $name";
    }
?>
```

១.៤.៧.៧ Array

Array គឺជា variable ពិសេសដែលអនុញ្ញាតអ្នក ផ្ទុកនូវតម្លៃបានច្រើននៅក្នុង variable មួយ។ រាល់តម្លៃនីមួយៗត្រូវរក្សាទុកនៅក្នុង index របស់ array ដែលអាចជាលេខឬតួអក្សរ។ ជា default ធាតុរបស់ array ដែល index គឺចាប់ផ្តើមពីលេខ 0។

ក. Index array

គឺជា array ដែលមាន index ជាលេខហើយចាប់ផ្តើមពីលេខ 0។ តម្លៃរបស់ index របស់វាអាចជា number, string ហើយនិង object នៅក្នុង array។

ឧទាហរណ៍៖

```
<?php
    $people = array("សុខា", "សំ", "រតថានា");
    $echo "សួស្តី ".$people[0];
?>
```

2. Associative array

គឺជា array ដែលមាន index អាចជាអក្សរ។

ឧទាហរណ៍៖

```
<?php
```

```
$people = array("a" => "សុខា", "b" => "សំ", "c" => "វតថាណា");
```

```
$echo "សូមស្តី ".$people['a'];
```

```
?>
```

១.៥ ទ្រឹស្តីពាក់ព័ន្ធនឹង Database

Database គឺជាកន្លែងផ្ទុកទិន្នន័យដែលមានទំនាក់ទំនងគ្នា និងអាចរក្សាទុកនៅក្នុងកន្លែងមួយដើម្បីធ្វើការរៀបចំ សម្រាប់អ្នកប្រើប្រាស់អាចទាញយកទិន្នន័យទៅតាមតម្រូវការផ្សេងៗ។ Database ត្រូវបានគ្រប់គ្រង និងរៀបចំដោយប្រព័ន្ធគ្រប់គ្រង Database Model ហើយវាបង្កើតឡើងដើម្បីបំពេញតម្រូវការទៅតាមប្រព័ន្ធណាមួយដែលគេចង់បាន។

១.៥.១ DBMS

DBMS ដែលមានអក្សរពេញ Database Management System គឺសំដៅទៅលើ Application Software ដែលត្រូវ បានប្រើសំរាប់កំណត់ បង្កើត និងថែរក្សា Database ព្រមទាំងផ្តល់នូវការត្រួតពិនិត្យ ទៅលើដំណើរការរបស់ Database ។

Database Management System ដែលយើងអាចប្រើរួមមាន៖

- Microsoft Access
- My SQL
- SQL Server
- Oracle

១.៥.១.១ My SQL

MySQL គឺជា Relational Database Management System ហើយវាជាប្រភេទកម្មវិធី Open Source Database ដ៏ពេញនិយមមួយនៅលើពិភពលោក ដែលយើងអាចយកមកប្រើប្រាស់ដោយមិនចាំបាច់បង់ថ្លៃការប្រើប្រាស់នោះទេ។ វាត្រូវបានគេពេញនិយមយកទៅប្រើប្រាស់ជាមួយ Web Application។ លក្ខណៈពិសេសរបស់ MySQL មានដូចខាងក្រោម៖

- ផ្ទុកទិន្នន័យបានច្រើន
- អាចប្រើប្រាស់បានជាមួយគ្រប់ Operating System

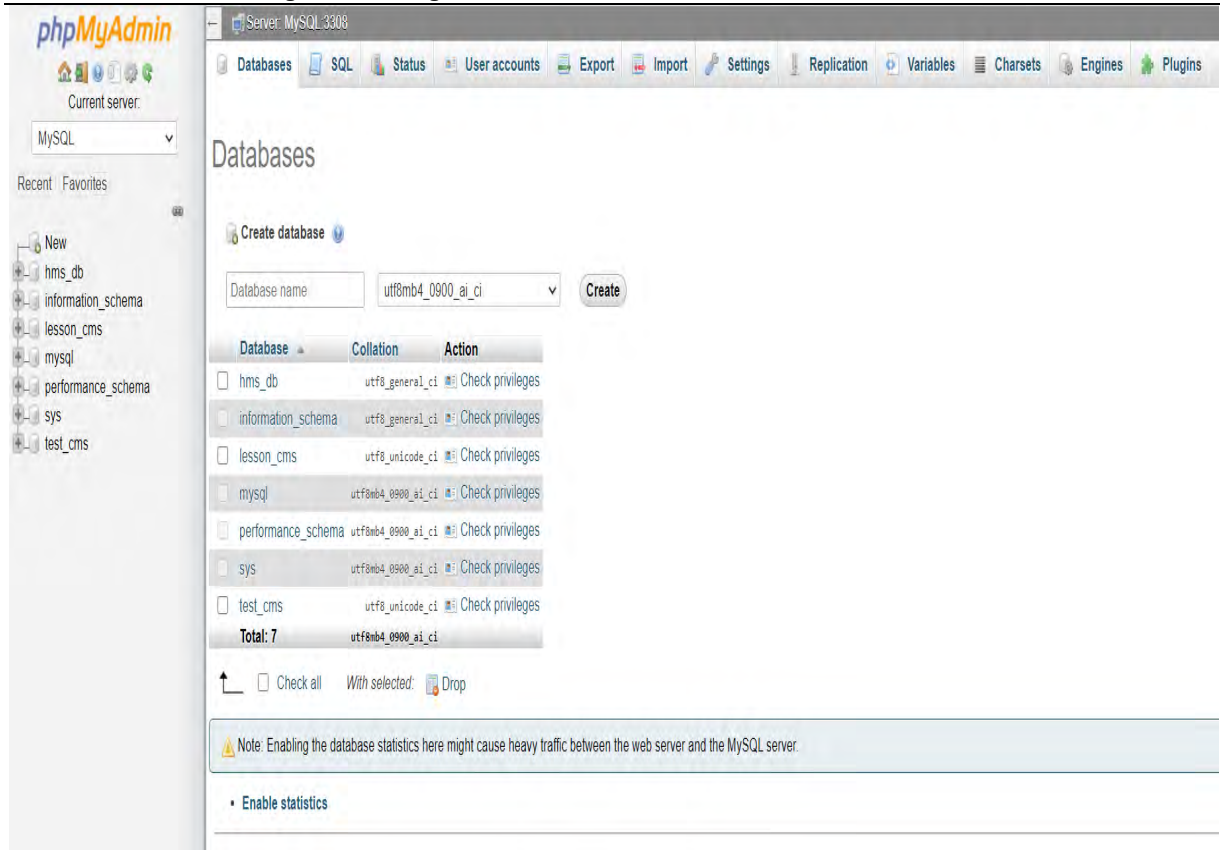
- ងាយស្រួលគ្រប់គ្រង និងកែតម្រូវពេលមានបញ្ហា
- អាចទាញយកទិន្នន័យមកប្រើប្រាស់តាមតម្រូវការ
- ងាយស្រួលដំណើរការ និងមានសុវត្ថិភាព



(រូបភាពទី ៣៦៖ រូបភាពតំណាងឲ្យ MySQL)

១.៥.១.២ វេយប៊ែរ PHP MyAdmin

PHP MyAdmin គឺជា Application សម្រាប់គ្រប់គ្រង និងត្រួតពិនិត្យ MySQL បានយ៉ាងងាយស្រួល។ PHP MyAdmin ផ្តល់នូវ GUI Interface សម្រាប់ប្រើប្រាស់ និងគ្រប់គ្រង MySQL Database ។ វាមានមុខងារជាច្រើន និងសំបូររបបដែលអនុញ្ញាតឲ្យអ្នកប្រើប្រាស់ ធ្វើការលុប និង បង្កើត Object ផ្សេងៗដូចជា Database, Table, Fields ជាដើម។ លើសពីនេះទៅទៀតយើងអាចធ្វើការ IMPORT និង EXPORT Database និងអាចប្រើប្រាស់ SQL Command ក្នុងនោះបានទៀតផង។



(រូបភាពទី ៣៧៖ Interface របស់ PHP MyAdmin)

១.៥.២ Data Analysis

Data Analysis គឺជាការវិភាគទិន្នន័យ Database កុំឲ្យទិន្នន័យមានលក្ខណៈស្ងួត និង ស្មុគស្មាញ។ វិធីសាស្ត្រដែលគេយកមកធ្វើការវិភាគនោះគឺ ការធ្វើ Normalization។ Normalization មានច្រើនវិធីសាស្ត្រដូចជា First Normal Form, Second Normal Form, Third Normal Form។ ខាងក្រោមនឹងបកស្រាយពីវិធីសាស្ត្រទាំង ៣ ដែលគេនិយមប្រើ និងមានលក្ខណៈដូចតទៅ៖

➤ First Normal Form

- Attribute ត្រូវមានឈ្មោះខុសគ្នា
- Attribute មួយមាន DataType តែមួយ និងមានតម្លៃតែមួយ
- គ្មាន Row ជាន់គ្នា
- គ្មាន Repeating Group

➤ Second Normal Form

- ដំបូងទម្រង់នេះត្រូវស្ថិតនៅក្នុង First Normal Form

- លុប Attribute ដែលអាស្រ័យដោយផ្នែកចេញ រួចដាក់វាចូល Table ថ្មីដែលមានន័យថាយើងបានបង្កើត Entity ឬ table ថ្មីដែលមាន Relationship ជាមួយ table ចាស់។

➤ Third Normal Form

- ដំបូងទម្រង់នេះត្រូវស្ថិតនៅក្នុង Second Normal Form
- លុប Attribute ដែលអាស្រ័យ table ថ្មី រួចដាក់ Relationship

១.៥.៣ Database Relationship

Database Relationship គឺជាការភ្ជាប់ទំនាក់ទំនង រវាង Entity មួយ ទៅ Entity ផ្សេងទៀត។ Relationship ចែកចេញជាបីប្រភេទគឺ One to One, One to Many, Many to Many។ ខាងក្រោមនេះជាតារាងពិពណ៌នា៖

Type of relationship	Description
One-to-one	Both tables can have only one record on each side of the relationship. Each primary key value relates to none or only one record in the related table. Most one-to-one relationships are forced by business rules and do not flow naturally from the data. Without such a rule, you can typically combine both tables without breaking any normalization rules.
One-to-many	The primary key table contains only one record that relates to none, one, or many records in the related table.
Many-to-many	Each record in both tables can relate to none or any number of records in the other table. These relationships require a third table, called an associate or linking table, because relational systems cannot directly accommodate the relationship.

(រូបភាពទី ៣៨៖ តារាងពិពណ៌នា Database Relationship)

ជំពូកទី ២

ដំណើរការនៃការស្រាវជ្រាវ

២.១ នីតិសាស្ត្រស្រាវជ្រាវ

២.១.១ ការពិភាក្សាដើម្បីជ្រើសរើសប្រធានបទ

បន្ទាប់ពីសម្រេចចិត្តជ្រើសរើសការសរសេរសារណា ដើម្បីបញ្ចប់ការសិក្សារួចមក យើងខ្ញុំបានបង្កើតក្រុមដែលមានសមាជិក ២ រូប ហើយក៏បានពិភាក្សាគ្នាដើម្បីធ្វើការជ្រើសរើសប្រធានបទដើម្បីធ្វើការស្រាវជ្រាវ និងចងក្រងជាសៀវភៅសារណាបញ្ចប់ការសិក្សា។ ការជ្រើសរើសប្រធានបទមកសរសេរយើងត្រូវជ្រើសប្រធានបទឲ្យសមស្របទៅនឹងសមត្ថភាព ចំណេះ ជំនាញ បទពិសោធន៍ និងរយៈពេលដើម្បីជៀសវាងជ្រើសរើសប្រធានដែលធំហួសសមត្ថភាព។ នេះជាជំហានដំបូងដែលមានសារៈសំខាន់បំផុតក្នុងការចាប់ផ្តើមសាងគម្រោងមួយនេះឡើងដើម្បីឲ្យយើងកំណត់បាននូវគោលដៅមួយដែលច្បាស់លាស់ និងត្រឹមត្រូវ។

២.១.២ ការជ្រើសរើសសាស្ត្រាចារ្យណែនាំគម្រោង

បន្ទាប់ពីធ្វើការសម្រេចចិត្តក្នុងការជ្រើសរើសប្រធានបទរួចមកក្រុមយើងខ្ញុំបានធ្វើការស្នើរសុំលោកសាស្ត្រាចារ្យ បណ្ឌិត ទិន ហេង ដែលជាសាស្ត្រាចារ្យនៃសកវិទ្យាល័យភូមិន្ទនីតិសាស្ត្រ និងវិទ្យាសាស្ត្រសេដ្ឋកិច្ចដើម្បីធ្វើសាស្ត្រណែនាំ ដោយលោកគ្រូមានជំនាញទាក់ទងនឹងប្រធានបទដែលក្រុមយើងខ្ញុំបានជ្រើសរើស។ លើសពីនេះទៅទៀតការជ្រើសរើសសាស្ត្រណែនាំគឺមានសារៈសំខាន់ណាស់ដើម្បីសម្រេចគម្រោងឲ្យបានជោគជ័យ ព្រោះសាស្ត្រាចារ្យអាចផ្តល់ដំណោះស្រាយល្អៗ ក៏ដូចជា មតិល្អៗ និងជួយជ្រោមជ្រែងបញ្ហាដែលបានកើតឡើងក្នុងកំឡុងពេលសរសេរគម្រោង ។

២.១.៣ ការប្រមូលទិន្នន័យ

ការប្រមូលទិន្នន័យក្នុងការសរសេរសារណាគឺមានលក្ខណៈស្មើគ្នាព្រោះទិន្នន័យដែលទទួលបានគឺមានលក្ខណៈមិនច្បាស់លាស់ ខុសពីទិន្នន័យដែលទទួលបានពីការចុះកម្មសិក្សាតាមស្ថាប័ននានាដែលទិន្នន័យដែលទទួលបានមានលក្ខណៈច្បាស់លាស់ និង ពិតប្រាកដ ពីម្ចាស់អធិវកម្មផ្ទាល់។ ទោះជាយ៉ាងណាក៏ដោយដើម្បីបង្កើតកម្មវិធីមួយឲ្យមានប្រសិទ្ធភាព ក្រុមយើងខ្ញុំបានខិតខំប្រឹងប្រែង ព្យាយាមដើម្បីប្រមូលទិន្នន័យ និងព័ត៌មានដែលទាក់ទងនឹងសណ្ឋាគារ ឬ ផ្ទះសំណាក់ ហើយស្វែងយល់ពីតម្រូវការឲ្យបានច្បាស់លាស់ពីអ្នកប្រើប្រាស់ដើម្បីបង្កើតគេហទំព័រគ្រប់គ្រងសណ្ឋាគារមួយដែលឆ្លើយតបទៅនឹងតម្រូវការម្ចាស់អធិវកម្មគ្រប់គ្រងសណ្ឋាគារ ឬផ្ទះសំណាក់ និងដើម្បីការងារសរសេរសារណាបញ្ចប់ការសិក្សាផងដែរ ។

❖ យុទ្ធសាស្ត្រនៃការប្រមូលទិន្នន័យ៖

- ព្យាយាមស្វែងរកទិន្នន័យតាមប្រព័ន្ធអ៊ីនធឺណិតដូចជា Social Network, Google និងវេបសាយមួយចំនួន

- សាកសួរព័ត៌មានមិត្តភក្តិដែលជាបុគ្គលិកផ្នែក Receptionist នៅក្នុងសណ្ឋាគារ

- សាកសួរអតិថិជនដែលធ្លាប់ចូលស្នាក់នៅសណ្ឋាគារ

២.១.៤ ការកំណត់តម្រូវការព័ត៌មាន

ជាក់ស្តែងបន្ទាប់ពីការប្រមូលទិន្នន័យរួចរាល់ហើយ ក្រុមយើងខ្ញុំបានកត់ត្រានូវចំណុចសំខាន់ៗរាល់ព័ត៌មានដែលទទួលបាន ហើយយកមកពិចារណា និងធ្វើការកំណត់តម្រូវការដែលច្បាស់លាស់មួយដើម្បីឲ្យសមស្របទៅនឹងចំណង់ចំណូលចិត្តរបស់អ្នកប្រើប្រាស់ និងតម្រូវការដែលអ្នកប្រើប្រាស់ចង់បាន។

២.១.៥ ការរៀបចំរចនាសម្ព័ន្ធនៃប្រព័ន្ធ

ជំហាននេះយើងបានធ្វើការរចនា User Interface ឲ្យសមស្របទៅនឹងតម្រូវការដែលបានកំណត់រួច ដែលធ្វើឃើញណាឲ្យមានភាពងាយស្រួលដល់អ្នកប្រើប្រាស់ និងឆាប់យល់ពីរបៀបប្រើប្រាស់។

ចំពោះ User Interface យើងខ្ញុំបានជ្រើសរើសភាសា Programming និងបច្ចេកវិទ្យាមួយចំនួនមកប្រើប្រាស់ដែលមានលក្ខណៈ Dynamic ដែលបង្កើនភាពស្រស់ស្អាត និងភាពទាក់ទាញផងដែរដូចជា Bootstrap, CSS, Java Script និង JQuery ជាដើម។

២.១.៦ ការសរសេរកូដ

ក្នុងដំណាក់កាលនេះគឺយើងធ្វើការសរសេរកូដដើម្បីធ្វើការបញ្ជាទៅលើ Interface ដែលយើងបាន Design នោះឲ្យធ្វើការទៅតាមអ្វីដែលយើងចង់បាន និងសរសេរកូដដាក់លក្ខខណ្ឌមួយចំនួនផងដែរ ដែលយើងអាចនិយាយបានថាគឺសរសេរកូដដើម្បីគ្រប់គ្រង Interface ទាំងមូលតែម្តង។

២.១.៧ ការពិសោធន៍សាកល្បងនិងថែរក្សាប្រព័ន្ធ

នេះជាដំណាក់កាល ដែលធ្វើការសាកល្បងលើការប្រើប្រាស់ប្រព័ន្ធ ដើម្បីធ្វើការស្វែងរកកំហុស និងបញ្ហាផ្សេងៗ ដែលអាចកើតឡើងដោយចៃដន្យ ឬនឹកស្មានមិនដល់ បើសិនមានកំហុសដែលបានកើតឡើង ឬក៏ចំនុចខ្វះខាត យើងត្រូវត្រលប់មកពិនិត្យការសរសេរកូដសារជាថ្មីដើម្បីធ្វើការកែតម្រូវឡើងវិញ។ ក្នុងការសាកល្បងនេះ គឺក្រុមយើងខ្ញុំសាកល្បងប្រើប្រាស់ដោយផ្ទាល់ ។

កំហុសដែលបានរកឃើញពីការសាកល្បងមានដូចជា៖

- ទាក់ទងទៅនឹង payment
- ការធ្វើរបាយការណ៍
- ទាក់ទងនឹង User permission

២.២ គម្រោងពេលវេលានៃការស្រាវជ្រាវ

ការគ្រប់ពេលវេលាគឺមានសារៈសំខាន់ខ្លាំងណាស់ មិនថាការរស់នៅប្រចាំថ្ងៃ ឬធ្វើការងារ កិច្ចការអ្វីឡើយបើយើងមិនបានគ្រប់គ្រង ឬកំណត់ពេលវេលាឲ្យបានច្បាស់ទេនឹងធ្វើឲ្យមានហានិភ័យកើតឡើង។ ដោយឡែកសម្រាប់ការកសាងគម្រោងនេះ ក្រុមយើងខ្ញុំបានបែកចែកពេលវេលាដើម្បីអនុវត្តន៍ដំណាក់កាលនីមួយៗ បានយ៉ាងច្បាស់លាស់។ ខាងក្រោមជាតារាងបែកចែកពេលវេលាដែលបានកំណត់ក្នុងការបង្កើតគម្រោងនេះឡើង ៖

❖ លេខសម្គាល់ដំណាក់កាល

១ ការស្រាវជ្រាវស្វែងរកទិន្នន័យ

២ ការវិភាគតម្រូវការ និង វិភាគ Database

៣ ការធ្វើគម្រោងសម្រាប់ Interface និង Database

៤ ដំណើរការសរសេរកូដ

៥ ដំណើរការសាកល្បង

៦ ការសរសេរសារណា

៧ ត្រួតពិនិត្យសារណាឡើងវិញ

ដំណាក់កាល	មេសា				ឧសភា				មិថុនា				កក្កដា			
	១	២	៣	៤	១	២	៣	៤	១	២	៣	៤	១	២	៣	៤
១																
២																
៣																
៤																
៥																
៦																
៧																

(រូបភាពទី ៣៩៖ តារាងគម្រោងពេលវេលានៃការស្រាវជ្រាវ)

ជំពូកទី ៣

ការអនុវត្តទៅលើគេហទំព័រ

៣.១ ការវិភាគបម្រើបម្រាស់ទិន្នន័យនៃប្រព័ន្ធ

យោងទៅតាមការដែលបានស្រាវជ្រាវកន្លងមកនេះ តាមលទ្ធផលដែលបានវិភាគលើគម្រោង ដោយការធ្វើបទសម្ភាសន៍ជាមួយសាស្ត្រាចារ្យដែលមានបទពិសោធន៍ និងការវិភាគផ្ទាល់លើសារណា ដែលអ្នកស្រាវជ្រាវខ្លួនឯងបានអនុវត្តន៍ក៏ទទួលបានលទ្ធផលដែលគួរជាទីទុកចិត្តបានថាជាទិន្នន័យមួយដែលអាចដាក់អោយប្រើប្រាស់បាន។ ចំពោះលទ្ធផលនៃការវិភាគទិន្នន័យសំខាន់ៗ គឺត្រូវរៀបចំឡើងដូចការដាក់បង្ហាញខាងក្រោម៖

❖ users: ផ្ទុកទិន្នន័យរបស់ Account User

Field Name	Data Type	Null	Primary Key
id	int(11)	No	Yes
group_id	int(11)	No	No
name	varchar(50)	No	No
email	varchar(100)	No	No
password	varchar(250)	No	No
created_at	timestamp	Yes	No
updated_at	timestamp	Yes	No

❖ group_permissions: ផ្ទុកទិន្នន័យរបស់ក្រុម User

Field Name	Data Type	Null	Primary Key
id	int(11)	No	Yes
name	varchar(50)	No	No
created_at	timestamp	Yes	No
updated_at	timestamp	Yes	No

❖ permissions: ផ្ទុកទិន្នន័យសិទ្ធិរបស់ក្រុម User ក្នុងការប្រើប្រាស់កម្មវិធី

Field Name	Data Type	Null	Primary Key
id	int(11)	No	Yes
group_id	int(11)	No	No
name	varchar(50)	No	No
created_at	timestamp	Yes	No
updated_at	timestamp	Yes	No

❖ guests: ផ្ទុកទិន្នន័យរបស់ភ្ញៀវ

Field Name	Data Type	Null	Primary Key
id	int(11)	No	Yes
name	varchar(50)	No	No
gender	varchar(20)	Yes	No
pob	varchar(255)	Yes	No
dob	date	Yes	No
nationality	varchar(100)	Yes	No
address	varchar(255)	Yes	No
phone	varchar(100)	Yes	No
email	varchar(100)	Yes	No
created_by	int(11)	No	No
updated_by	int(11)	Yes	No
created_at	timestamp	Yes	No
updated_at	timestamp	Yes	No

❖ rooms: ផ្ទុកទិន្នន័យរបស់បន្ទប់នីមួយៗ

Field Name	Data Type	Null	Primary Key
id	int(11)	No	Yes
name	varchar(50)	No	No
room_type_id	int(11)	No	No
floor_id	int(11)	No	No
description	varchar(255)	Yes	No
created_by	int(11)	No	No
updated_by	int(11)	Yes	No
deleted_by	int(11)	Yes	No
created_at	timestamp	Yes	No
updated_at	timestamp	Yes	No
deleted_at	datetime	Yes	No

❖ room_types: ផ្ទុកទិន្នន័យរបស់ប្រភេទបន្ទប់

Field Name	Data Type	Null	Primary Key
id	int(11)	No	Yes
name	varchar(50)	No	No
is_vip	int(1)	No	No
description	varchar(255)	Yes	No
created_by	int(11)	No	No
updated_by	int(11)	Yes	No
deleted_by	int(11)	Yes	No
created_at	timestamp	Yes	No
updated_at	timestamp	Yes	No
deleted_at	datetime	Yes	No

❖ floors: ផ្នែកទិន្នន័យរបស់ជាន់នីមួយៗរបស់បន្ទប់

Field Name	Data Type	Null	Primary Key
id	int(11)	No	Yes
name	varchar(50)	No	No
description	varchar(255)	Yes	No
created_by	int(11)	No	No
updated_by	int(11)	Yes	No
deleted_by	int(11)	Yes	No
created_at	timestamp	Yes	No
updated_at	timestamp	Yes	No
deleted_at	datetime	Yes	No

❖ room_transactions: ផ្នែកទិន្នន័យគ្រប់ប្រតិបត្តិការរបស់បន្ទប់

Field Name	Data Type	Null	Primary Key
id	int(11)	No	Yes
invoice_id	int(11)	No	No
guest_id	int(11)	No	No
room_id	int(11)	No	No
staying_type_id	int(11)	No	No
room_type_id	int(11)	No	No
from_date	datetime	No	No
to_date	datetime	No	No
duration	int(11)	No	No
discount_percent	decimal(6,2)	Yes	No
discount_amount	decimal(12,2)	Yes	No
amount	decimal(12,2)	No	No

grand_total	decimal(12,2)	No	No
status	varchar(20)	No	No
created_by	int(11)	No	No
updated_by	int(11)	Yes	No
deleted_by	int(11)	Yes	No
created_at	timestamp	Yes	No
updated_at	timestamp	Yes	No
deleted_at	datetime	Yes	No

❖ invoices: ផ្ទុកទិន្នន័យវិក័យបត្រសម្រាប់ធ្វើការទូទាត់ប្រាក់

Field Name	Data Type	Null	Primary Key
id	int(11)	No	Yes
guest_id	int(11)	No	No
reference	varchar(50)	Yes	No
discount_percent	decimal(6,2)	Yes	No
discount_amount	decimal(12,2)	Yes	No
amount	decimal(12,2)	No	No
grand_total	decimal(12,2)	No	No
status	varchar(20)	No	No
description	varchar(255)	Yes	No
paid	decimal(12,2)	Yes	No
refund	decimal(12,2)	Yes	No
payment_status	varchar(50)	No	No
created_by	int(11)	No	No
updated_by	int(11)	Yes	No
deleted_by	int(11)	Yes	No

សាកលវិទ្យាល័យភូមិន្ទនីតិសាស្ត្រ និងវិទ្យាសាស្ត្រសេដ្ឋកិច្ច

បរិញ្ញាបត្រសេដ្ឋកិច្ចព័ត៌មានវិទ្យា

created_at	timestamp	Yes	No
updated_at	timestamp	Yes	No
deleted_at	datetime	Yes	No

❖ payments: ផ្ទុកទិន្នន័យដែលបានទូទាត់សាច់ប្រាក់ជាមួយភ្ញៀវ

Field Name	Data Type	Null	Primary Key
id	int(11)	No	Yes
invoice_id	int(11)	No	No
guest_id	int(11)	No	No
amount	decimal(12,2)	No	No
description	varchar(255)	Yes	No
created_by	int(11)	No	No
updated_by	int(11)	Yes	No
deleted_by	int(11)	Yes	No
created_at	timestamp	Yes	No
updated_at	timestamp	Yes	No
deleted_at	datetime	Yes	No

❖ expends: ផ្ទុកទិន្នន័យការចំណាយផ្សេងៗ

Field Name	Data Type	Null	Primary Key
id	int(11)	No	Yes
reference	varchar(50)	Yes	No
name	varchar(100)	No	No
expend_type_id	int(11)	No	No
amount	decimal(12,2)	No	No
description	varchar(255)	Yes	No
created_by	int(11)	No	No
updated_by	int(11)	Yes	No
deleted_by	int(11)	Yes	No

created_at	Timestamp	Yes	No
updated_at	Timestamp	Yes	No
deleted_at	Datetime	Yes	No

❖ expend_types: ផ្ទុកទិន្នន័យប្រភេទនៃការចំណាយ

Field Name	Data Type	Null	Primary Key
id	int(11)	No	Yes
name	varchar(100)	No	No
amount	decimal(12,2)	No	No
description	varchar(255)	Yes	No
created_by	int(11)	No	No
updated_by	int(11)	Yes	No
deleted_by	int(11)	Yes	No
created_at	Timestamp	Yes	No
updated_at	Timestamp	Yes	No
deleted_at	Datetime	Yes	No

❖ transactions: ផ្ទុកទិន្នន័យរាល់ប្រតិបត្តិការដែលទាក់ទងនឹងសាច់ប្រាក់

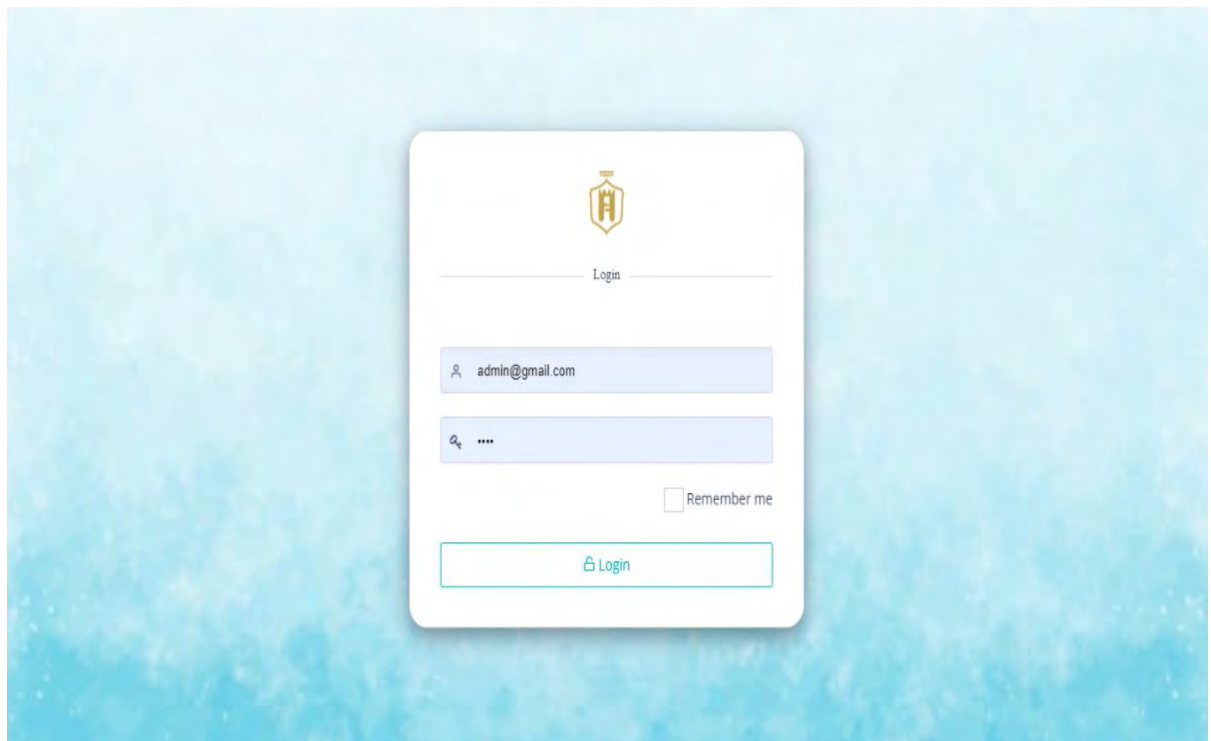
Field Name	Data Type	Null	Primary Key
id	int(11)	No	Yes
invoice_id	int(11)	Yes	No
guest_id	int(11)	Yes	No
payment_id	int(11)	Yes	No
expend_id	int(11)	Yes	No
expend_type_id	int(11)	Yes	No
amount	decimal(12,2)	No	No
currency_dollar	decimal(12,2)	No	No

currency_riel	decimal(12,2)	No	No
description	varchar(255)	Yes	No
created_by	int(11)	No	No
created_at	Timestamp	Yes	No

៣.២ ផ្ទាំង Interface នៃប្រព័ន្ធ

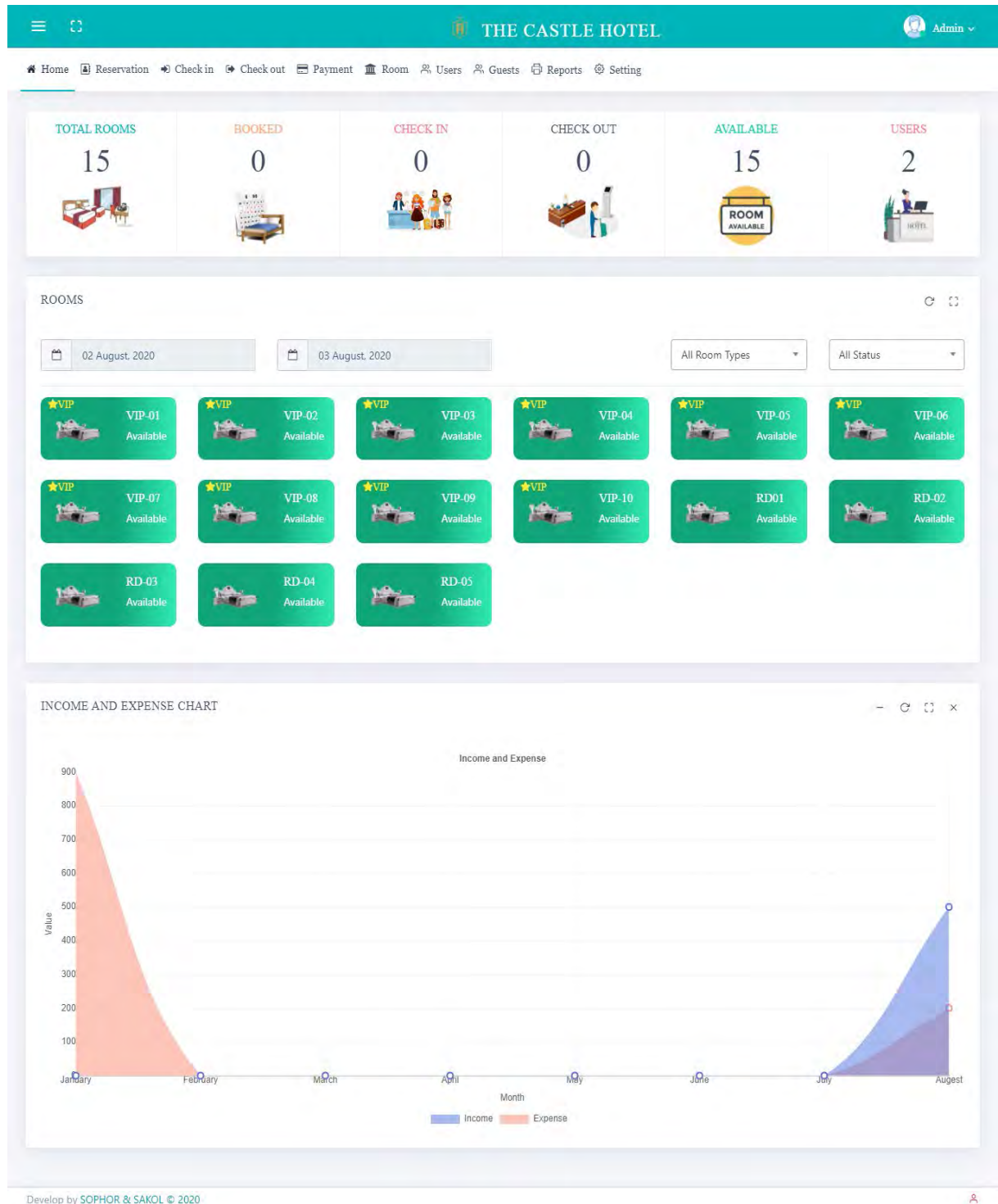
ដើម្បីងាយស្រួលដល់អ្នកប្រើប្រាស់ ក្នុងការប្រើកម្មវិធី កម្មវិធីនេះត្រូវបានរចនាទៅតាម តម្រូវការជាក់ស្តែង ដែលផ្តល់ភាពងាយស្រួលដល់អ្នកប្រើប្រាស់ តាមរយៈការបង្ហាញទម្រង់ និង បែងបទដូចខាងក្រោម៖

- Login Page: ជា form login ដែលអនុញ្ញាតអោយអ្នកប្រើប្រាស់ ត្រូវបំពេញព័ត៌មានដើម្បី ចូលប្រើប្រាស់កម្មវិធី។



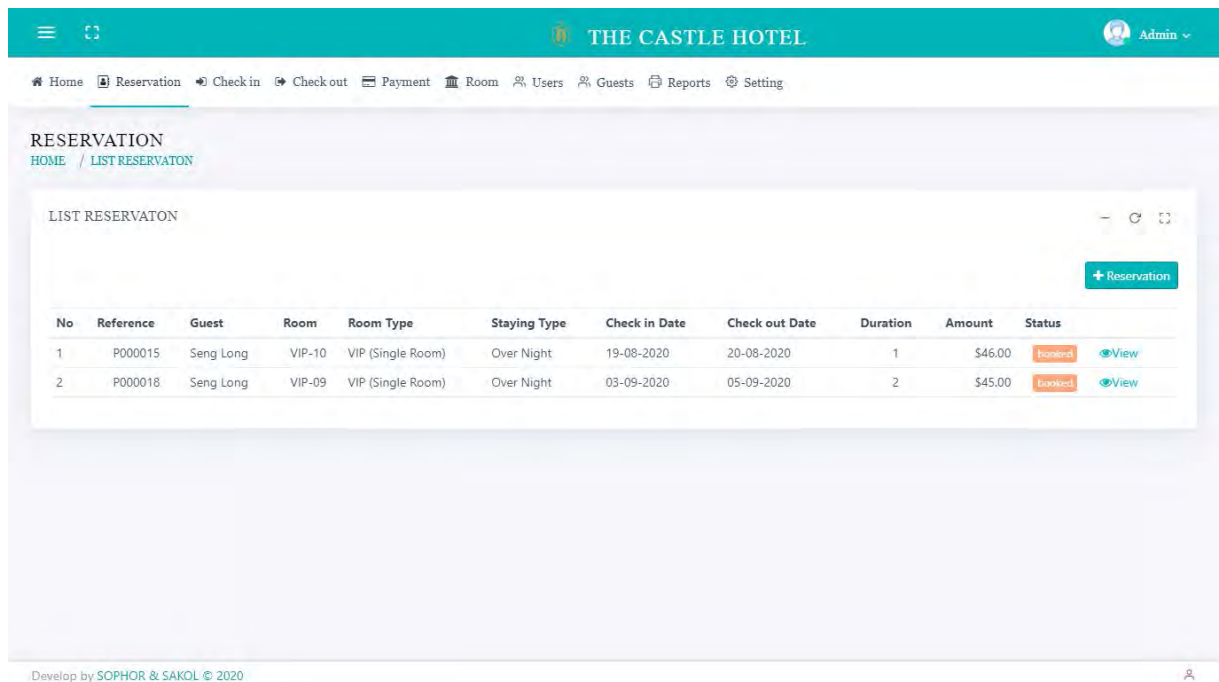
(រូបភាពទី៤០ ៖ បង្ហាញពី Login Page

• Home Page: ជាទំព័រដើមនៃកម្មវិធីសម្រាប់បង្ហាញពី ចំនួនបន្ទប់ ស្ថានភាពបន្ទប់ ចំនួនអ្នកប្រើប្រស់ ហើយនិងតារាងចំណូលចំណាយប្រចាំខែ។

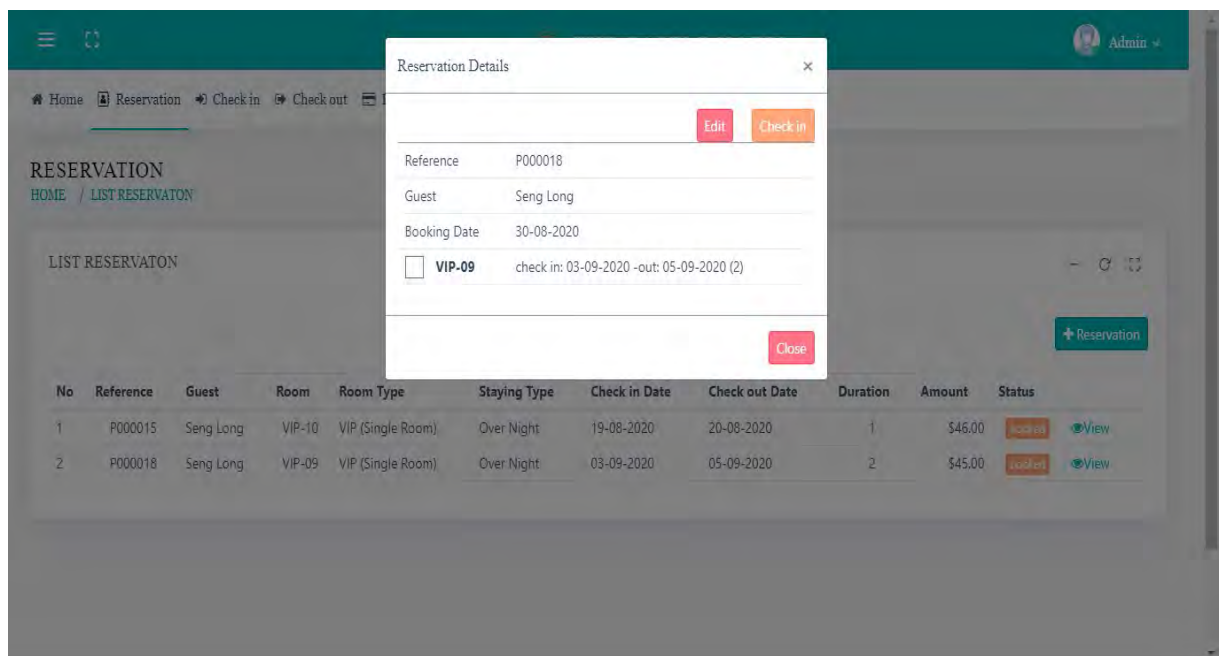


(រូបភាពទី ៤១៖ បង្ហាញពី Home Page

- Reservations: ជាផ្ទាំងដែលបង្ហាញពីបន្ទប់ដែលត្រូវបានកក់ ហើយយើងអាចធ្វើការ Add ,Cancel, Edit Reservation បាន។



(រូបភាពទី៤២ ៖ បង្ហាញពី Reservation Page



(រូបភាព ទី៤៣៖ បង្ហាញពី Reservation Detail

• Add Reservation: ជា Form សម្រាប់បញ្ចូលព័ត៌មានពីការកក់បន្ទប់។

ADD RESERVATION

Guest: Staying Type: Duration:

Checkin Date: Checkout Date:

Room: Amount: Discount:

No	Room	Room Type	Staying Type	Duration	Checkin	Checkout	Amount	Discount	Total	Function
Sub Total									\$0	
Discount %									0	
Deposit									0	
Grand Total									\$0	

Develop by SOPHOR & SAKOL © 2020

(រូបភាព ទី៤៤ ៖ បង្ហាញពី Form add Reservation

• Menu Checkin: ជាផ្ទាំងដែលបង្ហាញពីបន្ទប់ដែលភ្ញៀវបានកំពុងស្នាក់នៅ ហើយអាចចុចមើលព័ត៌មានលំអិតអំពីការ Check-In

CHECK IN

HOME / LIST CHECK IN

LIST CHECK IN

No	Reference	Guest	Room	Room Type	Staying Type	Check in Date	Check out Date	Duration	Amount	Status
1	P000016	Reaksa	RD01	Double Bed	Over Night	25-08-2020	26-08-2020	1	\$40.00	check in View
2	P000017	Reaksa	VIP-01	VIP (Single Room)	Over Night	25-08-2020	26-08-2020	1	\$40.00	check in View

Develop by SOPHOR & SAKOL © 2020

(រូបភាព ទី៤៥ ៖ បង្ហាញពី check-in page

- Edit Checkin: ជា Form សម្រាប់កែប្រែព័ត៌មាននៃការស្នាក់នៅ។

EDIT CHECK IN

Guest: 010 535 5662 | Reaksa

Staying Type: Select Staying Type

Duration: duration

Checkin Date:

Checkout Date:

Room:

Amount: 40.00

Room Type: Select Room Type

Discount: % discount

+ Add

No	Room	Room Type	Staying Type	Duration	Checkin	Checkout	Amount	Discount	Total	Function
1	RD01	Double Bed	Over Night	1	25-08-2020	26-08-2020	\$40	0%	\$40	remove
Sub Total									\$40	
Deposit									0	
Grand Total									\$40	

[Cancel](#) [Save](#)

(រូបភាព ទី៤៦ ៖ បង្ហាញពី form កែប្រែព័ត៌មាន Check-in)

- Checkout: ជាផ្ទាំងដែលបង្ហាញពីព័ត៌មានបន្ទប់ដែលភ្ញៀវបានចាកចេញ។

CHECK OUT

[HOME](#) / [LIST CHECK OUT](#)

LIST CHECK OUT

No	Reference	Guest	Room	Room Type	Staying Type	Check in Date	Check out Date	Duration	Amount	Status
1	P000012	Reaksa	VIP-01	VIP (Single Room)	Over Night	18-08-2020	19-08-2020	1	\$45.00	check out
2	P000013	Seng Long	VIP-02	VIP (Single Room)	Over Night	18-08-2020	19-08-2020	1	\$45.00	check out
3	P000014	Vandy	VIP-06	VIP (Single Room)	Over Night	18-08-2020	19-08-2020	1	\$45.00	check out

(រូបភាព ទី៤៧ ៖ បង្ហាញពីផ្ទាំង Check-Out)

- Payments: ជាផ្ទាំងដែលបង្ហាញពីបញ្ជីដែលភ្ញៀវត្រូវបង់ប្រាក់ និងបង់ប្រាក់ហើយ។

THE CASTLE HOTEL

Admin

Home Reservation Check in Check out Payment Room Users Guests Reports Setting

PAYMENTS

HOME / PAYMENTS

PAYMENTS

No	Reference	Guest	Amount	Discount	Grand Total	Paid	Status		
1	P000018	Seng Long	\$90.00	\$0.00	\$90.00	\$90.00	paid	Payment	Receipt
2	P000017	Reaksa	\$40.00	\$0.00	\$40.00	\$20.00	partial	Payment	Receipt
3	P000016	Reaksa	\$40.00	\$0.00	\$40.00	\$0.00	pending	Payment	Receipt
4	P000015	Seng Long	\$46.00	\$0.00	\$46.00	\$0.00	pending	Payment	Receipt
5	P000014	Vandy	\$45.00	\$0.00	\$45.00	\$0.00	pending	Payment	Receipt
6	P000013	Seng Long	\$45.00	\$0.00	\$45.00	\$0.00	pending	Payment	Receipt
7	P000012	Reaksa	\$45.00	\$0.00	\$45.00	\$0.00	pending	Payment	Receipt
8	P000011	Reaksa	\$25.00	\$0.00	\$25.00	\$0.00	pending	Payment	Receipt
9	P000010	Seng Long	\$60.00	\$0.00	\$58.80	\$0.00	pending	Payment	Receipt
10	P000009	Reaksa	\$25.00	\$0.75	\$24.25	\$0.00	pending	Payment	Receipt
11	P000008	Seng Long	\$40.00	\$0.00	\$39.60	\$0.00	pending	Payment	Receipt
12	P000007	Seng Long	\$36.00	\$5.00	\$31.00	\$0.00	pending	Payment	Receipt
13	P000006	Reaksa	\$25.00	\$0.50	\$24.50	\$0.00	pending	Payment	Receipt
14	P000005	Reaksa	\$0.00	\$0.00	\$0.00	\$0.00	pending	Payment	Receipt
15	P000004	Reaksa	\$32.00	\$0.32	\$31.68	\$31.68	paid	Payment	Receipt
16	P000003	Seng Long	\$30.00	\$0.30	\$29.70	\$29.70	paid	Payment	Receipt
17	P000002	Reaksa	\$35.00	\$0.00	\$69.30	\$69.30	paid	Payment	Receipt
18	P000001	Reaksa	\$80.00	\$0.00	\$149.00	\$149.00	paid	Payment	Receipt

Develop by SOPHOR & SAKOL © 2020

(រូបភាពទី៤៨ ៖ បង្ហាញពីផ្ទាំង Payment Page)

- Payment Form: ជា form ដែលត្រូវធ្វើការបង់ប្រាក់របស់ភ្ញៀវ។

Payment

Reference : P00002

Guest : Reaksa Kong

Paid amount : \$0.00

No	Room	Check in	Check out	Duration	Amount	Discount	Total
1	VIP-05	01-08-2020	02-08-2020	1	\$45.00		\$45.00
						Total	\$45.00

Amount

\$ amount

Payment

(រូបភាពទី ៤៩៖ បង្ហាញពីផ្ទាំង Form Payment)

- Users: បង្ហាញពីព័ត៌មានរបស់អ្នកប្រើប្រាស់ដែលមាននៅក្នុងកម្មវិធី។

THE CASTLE HOTEL

Admin

Home Reservation Check in Check out Payment Room Users Guests Reports Setting

USERS

HOME / LIST USERS

LIST USERS

Show 10 entries Search:

No	Name	Phone	Email	Address	User Name	Group	Status	Action
1	Admin Admin	010235645	admin@gmail.com	Phnom Penh	admin@gmail.com	ADMIN	Active	edit delete
2	Vandy Kao	010235678	vandykao@gmail.com	Phnom penh	vandykao@gmail.com	RECEPTIONIST	Active	edit delete

Showing 1 to 2 of 2 entries

Previous 1 Next

Develop by SOPHOR & SAKOL © 2020

(រូបភាពទី ៥០៖ បង្ហាញពីផ្ទាំង Users)

- Add User: ជា form ដែលបង្ហាញពីការបន្ថែមអ្នកប្រើប្រាស់ ដែលត្រូវបំពេញ Username, Password និង ព័ត៌មានអ្នកប្រើប្រាស់។

THE CASTLE HOTEL

Admin

Home Reservation Check in Check out Payment Room Users Guests Reports Setting

USERS

HOME / ADD USER

ADD USER

User Info

First Name
First Name

Last Name
Last Name

Phone
Phone

Email
Email

Address
Address

User Account

Group
OWNER

User Name
User Name

Password
Password

Confirm Password
Confirm Password

Cancel Save

Develop by SOPHOR & SAKOL © 2020

(រូបភាពទី៥១ ៖ បង្ហាញពីផ្ទាំងបញ្ចូលព័ត៌មានអ្នកប្រើប្រាស់)

- Edit User: ជា form ដែលបង្ហាញពីការកែប្រែអ្នកប្រើប្រាស់នៅក្នុងកម្មវិធី។

EDIT USER

User Info

First Name: Vandy

Last Name: Kao

Phone: 010235678

Email: vandykao@gmail.com

Address: Phnom penh

User Account

Group: RECEPTIONIST

User Name: vandykao@gmail.com

Password: Password

Confirm Password: Confirm Password

Cancel Save

Develop by SOPHOR & SAKOL © 2020

(រូបភាពទី៥២ ៖ បង្ហាញពីផ្ទាំងបញ្ចូលកែប្រែព័ត៌មានអ្នកប្រើប្រាស់)

សេចក្តីសន្និដ្ឋាន និងការផ្តល់អនុសាសន៍

១. សេចក្តីសន្និដ្ឋាន

បន្ទាប់បានសិក្សាស្រាវជ្រាវអស់រយៈពេល ៥ ខែកន្លងមក ក្រុមយើងខ្ញុំបានបង្កើតប្រព័ន្ធ គ្រប់គ្រងសណ្ឋាគារបានដោយជោគជ័យ ដែលអាចឲ្យម្ចាស់អដីកម្មអាចយកទៅប្រើប្រាស់ក្នុងមុខ ជំនួញរបស់ខ្លួនបាន។ ស្របជាមួយភាពជោគជ័យមួយនេះ យើងខ្ញុំសង្ឃឹមយ៉ាងមុតមាំថា ប្រព័ន្ធ គ្រប់គ្រងមួយនេះនឹងអាចមានលទ្ធភាពសម្រួលដល់ការគ្រប់គ្រងសណ្ឋាគារ ក៏ដូចជាការលំបាក មួយចំនួន ឲ្យមានភាពកាន់តែប្រសើរឡើង ។ ទោះបីជាប្រព័ន្ធនេះបានជួយដល់ការគ្រប់គ្រងអដី កម្មរបស់អ្នកប្រើប្រាស់ក៏ពិតមែន តែដោយសារពេលវេលាមានកំណត់ និងចំណេះដឹងនៅមានកំរិត ដែលធ្វើឲ្យមានចំណុចខ្វះខាត មិនអាចបំពេញនូវសេចក្តីត្រូវការបានគ្រប់ជ្រុងជ្រោយនៅឡើយ នោះទេ ហេតុដូច្នេះយើងខ្ញុំទាំអស់គ្នានឹងរងចាំរាល់ការវិះគន់ដើម្បីស្ថាបនា ក្នុងគោលដៅធ្វើឲ្យ ប្រព័ន្ធគ្រប់គ្រងមួយនេះកាន់តែមានភាពប្រសើរឡើងថែមទៀត។

២. ការផ្តល់អនុសាសន៍

ដើម្បីឲ្យដំណើរការនៃប្រព័ន្ធគ្រប់គ្រងនេះកាន់តែល្អប្រសើរ និងមានគុណភាពខ្ពស់ជាងនេះ ទៅទៀត យើងខ្ញុំសូមចែករំលែក និងផ្តល់អនុសាសន៍មួយចំនួនដូចខាងក្រោម៖

- ❖ អ្នកប្រើប្រាស់ប្រព័ន្ធមួយនេះគួរតែស្វែងយល់ពីដំណើរការរបស់ System មុនពេល ប្រើប្រាស់ System ដើម្បីជៀសវាងមានបញ្ហានៅពេលប្រើប្រាស់។
- ❖ អ្នកប្រើប្រាស់អាចទាក់ទងមកយើងខ្ញុំនៅពេល ប្រព័ន្ធមានបញ្ហា ឬមានចន្លោះខ្វះ ខាតណាមួយ និង ចង់បន្ថែមមុខងារលើប្រព័ន្ធបន្ថែមទៀត ។

ឯកសារយោង

១.ការដកស្រង់ចេញពីសៀវភៅអេឡិចត្រូនិច

- HTML & CSS មជ្ឈមណ្ឌលព័ត៌មានវិទ្យា អ៊ិនធឺណែត ដោយលោកគ្រូ ហូ ម៉ូនី
- PHP & MySQL សាលាបច្ចេកទេសដុនបូស្កូ ដោយសាស្ត្រាចារ្យ ព្រាប សាន

១.ការដកស្រង់ចេញពីសៀវភៅអេឡិចត្រូនិច

- <https://www.javatpoint.com/>
- <https://existek.com/blog/sdlc-models/>
- <https://www.w3schools.com/>
- <https://laravel.com/>
- <https://laracasts.com/>
- <https://getbootstrap.com/>
- <http://www.antkh.com/tech/2510>
- <https://www.techfree.info/jquery-articles/>
- <https://en.wikipedia.org/wiki/Internet>
- <https://cambo-report.com/article/766>

❖ Controller

➤ RoomTransaction Controller

- Function Index

```
function index(Request $request,
$checkin=0){

    if($checkin==1){

        $title = 'Check in';

        $title_list = 'List Check in';

    }elseif($checkin==2){

        $title = 'Check out';

        $title_list = 'List Check out';

    }else{

        $title = 'Reservation';

        $title_list = 'List Reservaton';

    }

    $reservations = new
RoomTransaction();

    if($checkin==1){

        $reservations = $reservations-
>where('room_transactions.status', '=',
'stay_over');

    }

    elseif($checkin==2){

        $reservations = $reservations-
>where('room_transactions.status', '=',
'check_out');
```

```
    }

    else{

        $reservations = $reservations-
>where('room_transactions.status', '=',
'booked');

    }

    $reservations = $reservations-
>select(DB::raw('

        room_transactions.*,

        invoices.id as invoice_id,

        invoices.reference as
invoice_referent,

        guests.name as guest_name,

        rooms.name as room_name,

        room_types.name as
room_type_name,

        staying_types.name as
staying_type_name

    '));

    if(!empty($request->search)){

        $reservations = $reservations-
>where(function($query) use ($request){

            $query->where('invoices.name',
'LIKE', '%'.$request->search.'%');

            $query->where('guests.name',
'LIKE', '%'.$request->search.'%');
```

```

        $query->where('rooms.name',
'LIKE', '%'.$request->search.'%');

        $query-
>where('room_types.name', 'LIKE',
'%'.$request->search.'%');

    });

}

$reservations = $reservations-
>leftJoin('invoices', 'invoices.id', '=',
'room_transactions.invoice_id')

->leftJoin('guests', 'guests.id', '=',
'room_transactions.guest_id')

->leftJoin('rooms', 'rooms.id', '=',
'room_transactions.room_id')

->leftJoin('room_types',
'room_types.id', '=',
'room_transactions.room_type_id')

->leftJoin('staying_types',
'staying_types.id', '=',
'room_transactions.staying_type_id')

->get();

return
view('backend.reservations.index',
compact('reservations','checkin','title','title_lis
t'));

}

```

- Function Create

```

function create(Request $request,
$checkin=0){

    if($checkin==1){

        $title = 'Check in';

        $title_add = 'Add Check in';

        $status = 'stay_over';

    }else{

        $title = 'Reservation';

        $title_add = 'Add Reservaton';

        $status = 'booked';

    }

    if($request->method()=='GET'){

        $gs = Guest::get();

        $guests[null] = 'Select Guest';

        foreach ($gs as $value) {

            $guests[$value->id] = $value-
>phone.' | '.$value->name;

        }

        $st =
StayingType::where('over_datetime_type',
'<>', 'over_time')->get();

        $staying_types[] = 'Select Staying
Type';

        foreach ($st as $value) {

```

```

        $staying_types[$value->id] = $value->name;
    }

    $rtype = RoomType::get();

    $room_types[null] = 'Select Room Type';

    foreach ($rtype as $value) {

        $room_types[$value->id] = ucfirst($value->name);

    }

    $rooms=[];

    $discount_types['discount_percent']='percent(%)';

    $discount_types['discount_amount']='amount';

    return
    view('backend.reservations.create',
    compact('guests', 'staying_types',
    'room_types','rooms','discount_types','checkin', 'title', 'title_add'));

    }else if($request->method()=='POST'){

        $this->validate($request, [

            'guest' => 'required|numeric',

            'staying_type_ids.*' => 'required',

            'room_ids.*' => 'required',

```

```

            'room_type_ids.*' => 'required',

            'durations.*' => 'required|min:1',

            'checkins.*' => 'required',

            'amounts.*' =>

            'required|numeric|min:0',

            'discount_types.*' =>

            'nullable|max:100',

            'discounts.*' =>

            'nullable|numeric|min:0',

            'deposit' =>

            'nullable|numeric|min:0'

        ]);

        // dd($request->total_discount_type);

        $guest = $request->guest;

        $staying_type_ids = $request->staying_type_ids;

        $room_ids = $request->room_ids;

        $room_type_ids = $request->room_type_ids;

        $durations = $request->durations;

        $checkins = $request->checkins;

        $amounts = $request->amounts;

        $discount_types = $request->discount_types;

        $discounts = $request->discounts;

```

```

        $deposit = $request->deposit;

        foreach ($room_ids as $key =>
$value) {

            $d_checkin = date('Y-m-d',
strtotime($checkins[$key]));

            $d_checkout = date('Y-m-d',
strtotime($d_checkin.' +'. $durations[$key].'
day'));

            $roomBusies =
RoomTransaction::roomBusies($d_checkin,
$d_checkout);

            foreach ($roomBusies as $k =>
$v1) {

                if($value==$v1){

                    if($checkin==1){

                        return redirect()->back()-
>withErrors(['Room at row ' .($key+1).' can
not check in!']->withInput());

                    }else{

                        return redirect()->back()-
>withErrors(['Room at row ' .($key+1).' can
not booking!']->withInput());

                    }

                }

            }

            if($checkin==1){

```

```

                if($d_checkin > date('Y-m-d')){

                    return redirect()->back()-
>withErrors(['Room at row ' .($key+1).' can
not check in!']->withInput());

                }

            }

        }

        $invoice = Invoice::create([

            'guest_id' => $guest,

            'amount' => 0,

            'grand_total' => 0,

            'status' => $status,

            'created_by' => Auth::id()

        ]);

        $amount_total =0;

        $grand_total =0;

        $sub_total=0;

        foreach ($room_ids as $key =>
$value) {

            $amount_total +=

            $amounts[$key];

            $grand_total_item =

            $amounts[$key] * $durations[$key];

            $discount_amount =0;

            $discount_percent =null;

```

```

        if($discount_types[$key] ==
'discount_percent'){ $discount_percent =
$discounts[$key];

        $discount_amount=
$grand_total_item*($discounts[$key]/100);

    }else{

        $discount_amount
=$discounts[$key];

    }

    $grand_total_item-
=$discount_amount;

    $sub_total += $grand_total_item;

    $d_checkin = date('Y-m-d',
strtotime($checkins[$key]));

    $d_checkout = date('Y-m-d',
strtotime($d_checkin.' +'. $durations[$key].'
day'));

    RoomTransaction::create([

        'invoice_id' => $invoice->id,

        'guest_id' => $guest,

        'room_id' => $value,

        'staying_type_id' =>
$staying_type_ids[$key],

        'room_type_id' =>
$room_type_ids[$key],

        'from_date' => $d_checkin,

        'to_date' => $d_checkout,

```

```

        'duration' => $durations[$key],

        'discount_percent' =>
$discount_percent,

        'discount_amount' =>
$discount_amount,

        'amount' => $amounts[$key],

        'grand_total' =>
$grand_total_item,

        'status' => $status,

        'created_by' => Auth::id()

    ]);

}

$total_discount_percent = null;

$total_discount_amount = 0;

if($request-
>total_discount_type=='discount_percent'){

    $total_discount_percent =
$request->total_discount;

    $total_discount_amount =
$sub_total*($request->total_discount/100);

}else{

    $total_discount_amount =
$request->total_discount;

}

$grand_total = $sub_total-
$total_discount_amount;

```

```

        $invoice->amount = $sub_total;

        $invoice->deposit = $request-
>deposit*1;

        $invoice->payment_status =
'pending';

        $invoice-
>discount_percent=$total_discount_percent;

        $invoice->discount_amount =
$total_discount_amount;

        $invoice->grand_total =
$grand_total;

        $invoice->paid += $request-
>deposit*1;

        $invoice->description='Booking
Room';

        $invoice->save();

        if(!empty($request->deposit) &&
$request->deposit>0){

            $payment = Payment::create([

                'invoice_id' =>$invoice->id,

                'guest_id' => $guest,

                'amount' => $request-
>deposit*1,                'description' =>
$status.' room',

                'created_by' => Auth::id()

            ]);

            Transaction::create([

```

```

                'invoice_id' =>$invoice->id,

                'guest_id' => $guest,

                'payment_id' => $payment->id,

                'amount' => $request-
>deposit*1,

                'description' => $status.' room',

                'created_by' => Auth::id()

            ]);

        }

        Session::put('remove_item', 1);

        Session::flash("message",
__('label.add_success'));

        Session::flash("alert-class", 'bg-
success');

        if($checkin==1){

            return redirect()-
>route('payments');

        }else{

            return redirect()-
>route('reservations');

        }

    }

}

```

- Function Edit

```
function edit(Request $request, $id,
$checkin=0){

    if($checkin==1){

        $title = 'Check in';

        $title_edit = 'Edit Check in';

        $status ='stay_over';

        $description ='Check in Room';

    }else{

        $title = 'Reservation';

        $title_edit = 'Edit Reservaton';

        $status ='booked';

        $description ='Booking Room';

    }

    $invoice = Invoice::find($id);

    if(empty($invoice)){

        Session::flash("message", 'Not
Found!');

        Session::flash("alert-class", 'bg-
danger');

        return redirect()->back();

    }

    if($request->method()=='GET'){

        $gs = Guest::get();

        $guests[null] = 'Select Guest';
```

```
        foreach ($gs as $value) {

            $guests[$value->id] = $value-
>phone.' | '.$value->name;

        }

        $st =
StayingType::where('over_datetime_type',
'<>', 'over_time')->get();

        $staying_types[] = 'Select Staying
Type';

        foreach ($st as $value) {

            $staying_types[$value->id] =
$value->name;

        }

        $rtype = RoomType::get();

        $room_types[null] ='Select Room
Type';

        foreach ($rtype as $value) {

            $room_types[$value->id] =
ucfirst($value->name);

        }

        $rooms=[];

        $discount_types['discount_percent']='percen
t(%)';

        $discount_types['discount_amount']='amount
';
```

```

        $total_discount_amount = $invoice-
>discount_amount;

        $total_discount_type =
'discount_amount';

        if(!empty($invoice-
>discount_percent) || $invoice-
>discount_percent>0){

$total_discount_amount=$invoice-
>discount_percent;

        $total_discount_type =
'discount_percent';

    }

    $room_transactions =
RoomTransaction::where([

        ['invoice_id', '=', $id],

        ['status', '=', $status]

    ]->get());

    $rows=[];

    foreach ($room_transactions as
$key => $value) {

        $room = Room::find($value-
>room_id);

        $room_type =
RoomType::find($value->room_type_id);

```

```

        $staying_type =
StayingType::find($value->staying_type_id);

        $discount_amount = $value-
>discount_amount;

        $discount_type =
'discount_amount';

        if(empty($value-
>discount_percent) || $value-
>discount_percent<=0){

            $discount_amount=$value-
>discount_percent;

            $discount_type =
'discount_percent';

        }

        $rows[$key]=[

            'room_no' => $value->room_id,

            'room_name' => isset($room-
>name)?$room->name:",

            'room_type' => $value-
>room_type_id,

            'room_type_name' =>
isset($room_type->name)?$room_type-
>name:",

            'staying_type' => $value-
>staying_type_id,

            'staying_type_name' =>
isset($staying_type->name)?$staying_type-
>name:",

```

```

        'duration' => $value->duration,

        'date_from' => date('d-m-Y',
strtotime($value->from_date)),

        'date_to' => date('d-m-Y',
strtotime($value->to_date)),

        'amount' => $value->amount*1,

        'discount' =>
$discount_amount*1,

        'discount_type' =>
$discount_type

    ];

}

return
view('backend.reservations.edit',
compact('guests','rows', 'invoice',
'staying_types',
'room_types','rooms','discount_types','total_
discount_type', 'total_discount_amount',
'checkin', 'title','title_edit'));

}else if($request->method()=='POST'){

    $this->validate($request, [

        // 'guest' => 'required|numeric',

        'staying_type_ids.*' => 'required',

        'room_ids.*' => 'required',

        'room_type_ids.*' => 'required',

        'durations.*' => 'required|min:1',

        'checkins.*' => 'required',

```

```

        'amounts.*' =>

'required|numeric|min:0',

        'discount_types.*' =>

'nullable|max:100',

        'discounts.*' =>

'nullable|numeric|min:0',

        'deposit' =>

'nullable|numeric|min:0'

    ]);

    // dd($request->all());

    $guest = $request->guest;

    $staying_type_ids = $request-
>staying_type_ids;

    $room_ids = $request->room_ids;

    $room_type_ids = $request-
>room_type_ids;

    $durations = $request->durations;

    $checkins = $request->checkins;

    $amounts = $request->amounts;

    $discount_types = $request-
>discount_types;

    $discounts = $request->discounts;

    $deposit = $request->deposit;

    foreach ($room_ids as $key =>
$value) {

```

```

        $d_checkin = date('Y-m-d',
strtotime($checkins[$key]));

        $d_checkout = date('Y-m-d',
strtotime($d_checkin.'+'. $durations[$key].
day));

        $roomBusies =
RoomTransaction::roomBusies($d_checkin,
$d_checkout, $id);

        foreach ($roomBusies as $k =>
$value) {

            if($value==$v){

                return redirect()->back()-
>withErrors(['Room at row ' .($key+1).' can
not booking!'])->withInput();

            }

        }

        $room_transactions =
RoomTransaction::where([

            ['invoice_id', '=', $id],

            ['status', '=', $status]

        ]->update(['status'=> 'cancel']);

        $amount_total =0;

        $grand_total =0;

        $sub_total=0;

        foreach ($room_ids as $key =>
$value) {

```

```

        $amount_total +=
$amounts[$key];

        $grand_total_item =
$amounts[$key] * $durations[$key];

        $discount_amount =0;

        $discount_percent =null;

        if($discount_types[$key] ==
'discount_percent'){

            $discount_percent =
$discounts[$key];

            $discount_amount=
$grand_total_item*($discounts[$key]/100);

        }else{

            $discount_amount
=$discounts[$key];

        }

        $grand_total_item-
=$discount_amount;

        $sub_total += $grand_total_item;

        $d_checkin = date('Y-m-d',
strtotime($checkins[$key]));

        $d_checkout = date('Y-m-d',
strtotime($d_checkin.'+'. $durations[$key].
day));

        RoomTransaction::create([

            'invoice_id' => $invoice->id,

```

```

        'guest_id' => $invoice-
>guest_id,

        'room_id' => $value,

        'staying_type_id' =>
$staying_type_ids[$key],

        'room_type_id' =>
$room_type_ids[$key],

        'from_date' => $d_checkin,

        'to_date' => $d_checkout,

        'duration' => $durations[$key],

        'discount_percent' =>
$discount_percent,

        'discount_amount' =>
$discount_amount,

        'amount' => $amounts[$key],

        'grand_total' =>
$grand_total_item,

        'status' => $status,

        'created_by' => Auth::id()
    ]);
}

$total_discount_percent = null;

$total_discount_amount = 0;

if($request-
>total_discount_type=='discount_percent'){

```

```

        $total_discount_percent =
$request->total_discount;

        $total_discount_amount =
$sub_total*($request->total_discount/100);

    }else{

        $total_discount_amount =
$request->total_discount;

    }

    $grand_total = $sub_total-
$total_discount_amount;

    $invoice->amount = $amount_total;

    $old_deposit = $invoice->deposit;

    $invoice->deposit = $request-
>deposit*1;

    if($request->deposit*1 > 0){

        $invoice->payment_status =
'partial';

    }else{

        $invoice->payment_status =
'pending';

    }

    $invoice-
>discount_percent=$total_discount_percent;

    $invoice->discount_amount =
$total_discount_amount;

    $invoice->grand_total =
$grand_total;

```

```
$invoice->description= $status.'  
Room';
```

```
$invoice->save();
```

```
$payment = Payment::create([  
    'invoice_id' => $invoice->id,  
    'guest_id' => $invoice->guest_id,  
    'amount' => $old_deposit*(-1),  
    'description' => $description,  
    'created_by' => Auth::id()
```

```
]);
```

```
Transaction::create([  
    'invoice_id' => $invoice->id,  
    'guest_id' => $invoice->guest_id,  
    'payment_id' => $payment->id,  
    'amount' => $old_deposit*(-1),  
    'description' => $description,  
    'created_by' => Auth::id()
```

```
]);
```

```
$payment = Payment::create([  
    'invoice_id' => $invoice->id,  
    'guest_id' => $invoice->guest_id,  
    'amount' => $request->deposit*1,  
    'description' => $description,  
    'created_by' => Auth::id()
```

```
]);
```

```
Transaction::create([  
    'invoice_id' => $invoice->id,  
    'guest_id' => $invoice->guest_id,  
    'payment_id' => $payment->id,  
    'amount' => $request->deposit*1,  
    'description' => $description,  
    'created_by' => Auth::id()  
]);
```

```
Session::put('remove_item', 1);
```

```
Session::flash("message",  
__('label.updated_success'));
```

```
Session::flash("alert-class", 'bg-  
success');
```

```
if($checkin==1){  
    return redirect()-  
>route('checkins',['checkin'=>1]);
```

```
}else{
```

```
    return redirect()-  
>route('reservations');
```

```
}
```

```
}
```

```
}
```

- Function get date check out

```
function get_dateto(Request $request){

    $datefrom = date('Y-m-d',
strtotime($request->datefrom));

    return date('d F, Y',
strtotime($datefrom.'+ '.$request->duration.'
day'));

}
```

- Function get room

```
function get_rooms(Request $request){

    $from_date = isset($request-
>date_from)?$request->date_from:date('Y-
m-d');

    $to_date = date('Y-m-d',
strtotime($from_date.'+ '.$request-
>duration.' day'));

    // dd($to_date);

    $roomBusies =
RoomTransaction::roomBusies($from_date,
$to_date);

    $rooms =
Room::where('room_type_id', '=', $request-
>room_type_id);

    if(count($roomBusies)>0){

        // dd($roomBusies);

        $rooms = $rooms-
>whereNotIn('id',$roomBusies);
```

```
}
```

```
$rooms = $rooms->get();
```

```
$r_rooms[null]='-- Select Room --';
```

```
foreach ($rooms as $room) {
```

```
    $r_rooms[$room->id] =
ucfirst($room->name);
```

```
}
```

```
$select_option = Form::select('room',
$r_rooms,null, ['class' => 'select2-data-array
form-control', 'style' =>
'width:100%!important;', 'id' => 'room']);
```

```
return $select_option;
```

```
}
```

- Function get room price

```
function get_room_price(Request
$request){

    if(!$request->all())

        return 0;

    $roomfee = RoomFee::where([

        ['room_id', '=', $request-
>room_id],

        ['staying_type_id', '=',
$request->staying_type]

    ]->first();

    if($roomfee)
```

```

        return $roomfee-
>fee*$roomfee->count_number;

        return 0;

    }

    - Function Show

    public function show(Request $request){

        $data['html'] = "";

        if(!empty($request->id)){

            $invoice = Invoice::find($request-
>id);

            if(!empty($invoice)){

                $reservations =
RoomTransaction::select(DB::raw('

                    room_transactions.duration as
duration,

                    room_transactions.id as id,

DATE_FORMAT(room_transactions.from_d
ate, "%d-%m-%Y") as checkin_date,

DATE_FORMAT(room_transactions.to_date
, "%d-%m-%Y") as checkout_date,

                    rooms.name as room_name

                '))

            -

            >where('room_transactions.invoice_id', '=',
$invoice->id);

```

```

        $checkin = $request->checkin;

        if($checkin==1){

            $reservations = $reservations-
>where('room_transactions.status', '=',
'stay_over');

        }else{

            $reservations = $reservations-
>where('room_transactions.status', '=',
'booked');

        }

        $reservations = $reservations-
>leftJoin('rooms', 'rooms.id',
'room_transactions.room_id')

        ->get();

        $guest = Guest::find($invoice-
>guest_id);

        $data['html'] .=
".View::make('backend.reservations.view',
compact('reservations',
'invoice','guest','checkin'));

    }

}

return $data;

}

```

- Function checkin

```
public function checkin(Request $request,
$id){

    $data['message'] = 0;

    $this->validate($request,[

        'reservation.*' => 'required'

    ]);

    $reservation_ids = $request-
>reservation;

    if($request->checkin==1){

        foreach ($reservation_ids as $key
=> $value) {

            $transaction =
RoomTransaction::find($value);

            $t_date = date('Y-m-d',
strtotime($transaction->to_date));

            if($t_date < date('Y-m-d')){

                return $data;

            }

        }

    }else{

        foreach ($reservation_ids as $key
=> $value) {

            $transaction =
RoomTransaction::find($value);
```

```
        $t_date = date('Y-m-d',
strtotime($transaction->from_date));

        if($t_date > date('Y-m-d')){

            return $data;

        }

    }

    if(!empty($reservation_ids) &&
!empty($id)){

        if($request->checkin==1){

            $room_checked =
RoomTransaction::where([

                ['invoice_id', '=', $id],

                ['status', '=', 'stay_over']

            ])

            ->whereIn('id', $reservation_ids)

            ->update(['status' =>
'check_out']);

        }else{

            $room_checked =
RoomTransaction::where([

                ['invoice_id', '=', $id],

                ['status', '=', 'booked']

            ])

            ->whereIn('id', $reservation_ids)
```

```

        ->update(['status' =>
'stay_over']);

    }

    if(!empty($room_checked)){

        $data['message'] = 1;

    }

}

return $data;

}

```

➤ Room Controller

- Function Index

```

public function index(){

    $rooms = Room::get();

    return view('backend.rooms.index',
compact('rooms'));

}

```

- Function Create

```

public function create(){

    $roomType = RoomType::get();

    $floor = Floor::get();

    $rStatus = RoomStatus::get();

    $roomTypes = [];

    foreach($roomType as $type){

```

```

        $roomTypes [$type->id] =
ucfirst($type->name);

    }

    $floors = [];

    foreach($floor as $f){

        $floors[$f->id] = ucfirst($f->name);

    }

    $roomStatus = [];

    foreach($rStatus as $status){

        $roomStatus[$status->key_status] =
ucfirst($status->room_status);

    }

    return
view('backend.rooms.create',compact('room
Types', 'floors', 'roomStatus'));

}

public function store(Request $request){

    $user_id = Auth::user()->id;

    $this->validate($request, [

        'name' => 'required|string|max:255',

        'room_type' => 'required|integer',

        'floor' => 'required|integer',

        'description' =>

'nullable|string|max:255',

    ]);

```

```

        $room = Room::create([
            'name' => $request->name,
            'room_type_id' => $request-
>room_type,
            'floor_id' => $request->floor,
            'description' => $request-
>description,
            'created_by' => $user_id
        ]);

        if($room){
            Session::flash("message",
__('label.add_success'));

            Session::flash("alert-class", 'bg-
success');

        }else{

            Session::flash("message",
__('label.add_fail'));

            Session::flash("alert-class", 'bg-
danger');

        }

        return redirect()->route('room.create');
    }

```

- Function Edit

```

public function edit(Room $room){

    $roomType = RoomType::get();

    $floor = Floor::get();

    $roomTypes =[];

    foreach($roomType as $type){

        $roomTypes [$type->id] =
ucfirst($type->name);

    }

    $floors =[];

    foreach($floor as $f){

        $floors[$f->id] = ucfirst($f->name);

    }

    return view('backend.rooms.edit',
compact('room', 'roomTypes', 'floors'));

}

public function update(Room $room,
Request $request){

    $user_id = Auth::user()->id;

    $this->validate($request, [

        'name' => 'required|string|max:255',

        'room_type_id' => 'required|int',

        // 'floor' => 'required|int',

        'description' => 'string|max:255',

    ],[],[

```

```

        'room_type_id' => 'Room Type',

        'floor_id' => 'Floor',

    ]);

    $room->name = $request->name;

    $room->room_type_id = $request-
>room_type_id;

    $room->floor_id = $request->floor_id;

    $room->description = $request-
>description;

    $room->updated_by = $user_id;

    $room->save();

    if($room){

        Session::flash("message",
        __('label.updated_success'));

        Session::flash("alert-class", 'bg-
success');

    }else{

        Session::flash("message",
        __('label.updated_fail'));

        Session::flash("alert-class", 'bg-
danger');

    }

    return redirect()->route('room.index');

```

```

    }

➤ User Controller
    - Function Create

    public function create()

    {

        //

        $groups = [];

        $g_permission =
GroupPermission::select('id','name')->get();

        foreach($g_permission as $g){

            $groups[$g->id] = strtoupper($g-
>name);

        }

        return view("backend.staff.create",
compact('groups'));

    }

    public function store(Request $request)

    {

        $user_id= Auth::user()->id;

        $this->validate($request, [

            'first_name' => 'required|max:255',

            'last_name' => 'required|max:255',

            'phone'     => 'max:255',

            'email'     => 'unique:staff,email',

            'address'   => 'max:255',

```

```

        'user_name'      =>
'required|email|max:255|unique:users,email',

        'password'      =>
'required|min:4|max:255',

        'confirm_password' =>
'same:password',

        'group'      => 'required',
    ]);

    $user = User::create([

        'group_id' => $request->group,

        'name' => $request->first_name.'
'.$request->last_name,

        'email' => $request->user_name,

        'password' => bcrypt($request-
>password),

    ]);

    $staff = Staff::create([

        'user_id' => $user->id,

        'first_name' => $request-
>first_name,

        'last_name' => $request-
>last_name,

        'phone' => $request->phone,

        'address' => $request->address,

        'email' => $request->email,

        'status' => 1,

```

```

        'created_by' => $user_id,
    ]);

    if($user && $staff){

        Session::flash("message",
__('label.add_success'));

        Session::flash("alert-class", 'bg-
success');

    }else{

        Session::flash("message",
__('label.add_fail'));

        Session::flash("alert-class", 'bg-
danger');

    }

    return redirect()->route('staff.create');

}

```

- Function Edit

```

public function edit(Staff $staff)

{

    $groups = [];

    $g_permission =
GroupPermission::select('id','name')->get();

    foreach($g_permission as $g){

        $groups[$g->id] = strtoupper($g-
>name);

    }

```

```

        return view('backend.staff.edit',
compact('staff', 'groups'));

    }

    public function update(Request $request,
Staff $staff)

    {

        $owner_id = Auth::user()->id;

        $user_staff = User::where('id', $staff-
>user_id)->first();

        $this->validate($request, [

            'first_name' => 'required|max:255',

            'last_name' => 'required|max:255',

            'phone'     => 'max:255',

            'email'     => 'max:255',

            'address'   => 'max:255',

            'user_name' =>
'required|unique:users,email,'.$staff-
>user_id.',id',

        ]);

        $staff->first_name = $request-
>first_name;

        $staff->last_name = $request-
>last_name;

```

```

        $staff->phone = $request->phone;

        $staff->email = $request->email;

        $staff->address = $request->address;

        $staff->email = $request->email;

        $user_staff->email = $request-
>user_name;

        $user_staff->name = $request-
>first_name.' '.$request->last_name;

        if($request->password != ""){

            $this->validate($request, [

                'password'     =>
'required|min:4|max:255',

                'confirm_password' =>
'same:password',

            ]);

            $user_staff->password =
bcrypt($request->password);

        }

        $staff->save();

        $user_staff->save();

        if($staff){

            Session::flash("message",
__('label.updated_success'));

```

```

        Session::flash("alert-class", 'bg-
success');

    }else{

        Session::flash("message",
__('label.updated_fail'));

        Session::flash("alert-class", 'bg-
danger');

    }

    return redirect()->route('staff.index');

}

```

➤ Home Controller

- Function Index

```

public function index(Request $request)

{

    $from_date = isset($request-
>from_date)?$request->from_date:date('Y-
m-d');

    $to_date = isset($request-
>to_date)?$request->to_date:date('Y-m-d',
strtotime(date('Y-m-d').' +1 day'));

    $from_date = date('Y-m-d',
strtotime($from_date));

    $to_date = date('Y-m-d',
strtotime($to_date));

    $rooms = new Room();

    $rooms = $rooms

->select(DB::raw('

```

```

rooms.*,

room_types.is_vip as room_vip,

room_transactions.status as
room_status

'));

if(!empty($request->room_type)){

    $rooms = $rooms-
>where('room_types.id', '=', $request-
>room_type);

}

if(!empty($request->status)){

    if($request->status=='booked' ||
$request->status=='stay_over'){

        $rooms = $rooms-
>where('room_transactions.status', '=',
$request->status);

    }

}

$rooms = $rooms->join('room_types',
'room_types.id', '=', 'rooms.room_type_id')

->leftJoin('room_transactions',
function($query) use ($from_date, $to_date,
$request){

    $query-
>on('room_transactions.room_id', '=',
'rooms.id');

    if($request->status=='available'){

```

```

        $query-
>whereNotIn('room_transactions.status',['bo
oked', 'stay_over']);

    }

    $query-
>where(function($sub_query)
use($from_date, $to_date){

        $sub_query-
>whereBetween('from_date', [$from_date,
date('Y-m-d', strtotime($to_date.' - 1 day'))]);

        $sub_query-
>orWhereBetween('to_date', [date('Y-m-d',
strtotime($from_date.' + 1 day')), $to_date]);

    });

});

$rooms = $rooms->orderBy('rooms.id',
'ASC')->paginate(29);

$rtype = RoomType::get();

$room_types[null] = 'All Room Types';

foreach ($rtype as $value) {

    $room_types[$value->id] =
ucfirst($value->name);

}

$statuses[NULL] = 'All Status';

$statuses['available'] = 'Available';

$statuses['booked'] = 'Booked';

$statuses['stay_over'] = 'Staying';

```

```

        $from_date = date('d F, Y',
strtotime($from_date));

        $to_date = date('d F, Y',
strtotime($to_date));

        //count room

        $rbooked =
Room::where('room_transactions.status', '=',
'booked')

        ->leftJoin('room_transactions',
function($query) use ($from_date, $to_date,
$request){

            $query-
>on('room_transactions.room_id', '=',
'rooms.id');

            $query-
>where(function($sub_query)
use($from_date, $to_date){

                $sub_query-
>whereBetween('from_date', [date('Y-m-d'),
date('Y-m-d')]);

                $sub_query-
>orWhereBetween('to_date', [date('Y-m-d',
strtotime(date('Y-m-d').' + 1 day')),
$to_date]);

            });

        }->get()->count();

```

```

        $rstaying =
Room::where('room_transactions.status', '=',
'stay_over')

        ->leftJoin('room_transactions',
function($query) use ($from_date, $to_date,
$request){

        $query-
>on('room_transactions.room_id', '=',
'rooms.id');

        $query-
>where(function($sub_query)
use($from_date, $to_date){

        $sub_query-
>whereBetween('from_date', [date('Y-m-d'),
date('Y-m-d')]);

        $sub_query-
>orWhereBetween('to_date', [date('Y-m-d',
strtotime(date('Y-m-d').' + 1 day')),
$to_date]);

        });

    }->get()->count();

    $rtotal = Room::get()->count();

    $ravailable = $rtotal-$rbooked-
$rstaying;

    return view('backend.dashboard',
compact('rooms',
'room_types','statuses','from_date',

```

```

'to_date','request','rtotal','rbooked',
'rstaying','ravailable'));

```

```

}

```

➤ Guest Controller

- Function Index

```

public function index()

```

```

{

```

```

    $guests = Guest::get();

```

```

    return view("backend.guests.index",
compact('guests'));

```

```

}

```

- Function Create

```

public function create()

```

```

{

```

```

    $genders = [];

```

```

    $genders['male']="Male";

```

```

    $genders['female']="Female";

```

```

    return view("backend.guests.create",
compact('genders'));

```

```

}

```

```

public function store(Request $request)

```

```

{

```

```

    $user_id= Auth::user()->id;

```

```

    $this->validate($request, [

```

```

        'name' => 'required|max:255',

```

```

        'phone'    => 'max:255',
        'email'    => 'unique:guests,email',
        'address'  => 'max:255',
        'gender'   => 'required',
        'description' => 'max:255',
    ]);

    $guest = Guest::create([
        'name' => $request->name,
        'gender' => $request->gender,
        'dob' => date('Y-m-d'),
        'phone' => $request->phone,
        'address' => $request->address,
        'email' => $request->email,
        'description' => $request->description,
        'status' => 1,
        'created_by' => $user_id,
    ]);

    if($guest){
        Session::flash("message",
            __('label.add_success'));

        Session::flash("alert-class", 'bg-
            success');
    }else{

```

```

        Session::flash("message",
            __('label.add_fail'));

        Session::flash("alert-class", 'bg-
            danger');
    }

    return redirect()->route('guest.create');
}

```

- Function Edit

```

public function edit(Guest $guest)
{
    $genders = [];
    $genders['male'] = "Male";
    $genders['female'] = "Female";

    return view('backend.guests.edit',
        compact('guest', 'genders'));
}

public function update(Request $request,
    Guest $guest)
{
    $user_id = Auth::user()->id;

    $this->validate($request, [
        'name' => 'required|max:255',
        'phone'    => 'max:255',

```

```

        'email'    =>
'unique:guests,email,'.$guest->id.',id',

        'address'  => 'max:255',

        'gender'   => 'required',

        'description' => 'max:255',

    ]);

    $guest->name = $request->name;

    $guest->gender = $request->gender;

    $guest->dob = date('Y-m-d');

    $guest->phone = $request->phone;

    $guest->email = $request->email;

    $guest->address = $request-
>address;

    $guest->description = $request-
>description;

    $guest->updated_by = $user_id;

    $guest->save();

    if($guest){

        Session::flash("message",
__('label.updated_success'));

        Session::flash("alert-class", 'bg-
success');

    }else{

        Session::flash("message",
__('label.updated_fail'));

```

```

        Session::flash("alert-class", 'bg-
danger');

    }

    return redirect()->route('guest.index');

}

```

- Function Delete

```

public function destroy(Guest $guest)

{

    $user_id = Auth::user()->id;

    $guest->deleted_by = $user_id;

    $guest->save();

    $guest->delete();

    if($guest){

        Session::flash("message",
__('label.deleted_success'));

        Session::flash("alert-class", 'bg-
success');

    }else{

        Session::flash("message",
__('label.deleted_fail'));

        Session::flash("alert-class", 'bg-
danger');

    }
}

```

```

        return redirect()->route('guest.index');
    }

    ➤ Setting Controller
        - Function Group Permission

    public function groupPermissions(){

        $group_permissions =
        GroupPermission::get();

        return
        view('backend.permissions.group_index',
        compact('group_permissions'));

    }

        - Function Add Permission

    public function addPermissions($id){

        $group = $id;

        $group_name =
        GroupPermission::where('id' , $id)->first()-
        >name;

        $permission_data =
        GroupPermission::where('id', $id)->first()-
        >permission;

        $permissions =array();

        foreach($permission_data as $data){

            $permissions =
            array_merge($permissions, [$data->name
            => true]);

        }

```

```

        return
        view('backend.permissions.permission_form'
        , compact('permissions', 'group_name',
        'group'));

    }

        - Function Save Permission

    public function savePermissions(Request
    $request){

        $permissions = $request->permission;

        $group = $request->group;

        try {

            DB::beginTransaction();

            Permission::where('group_id',$group)-
            >delete();

            if($permissions !=null or
            $permissions !="){

                for($i=0;$i<=count($permissions)-1;$i++)

                    {

                        if
                        (strpos($permissions[$i], '||'))

                            {

```

```
$new_permission = explode("||",
$permissions[$i]);
```

```
foreach($new_permission as $row)
```

```
{
```

```
Permission::create([
```

```
'group_id' => $group,
```

```
'name' => $row
```

```
]);
```

```
}
```

```
}
```

```
else
```

```
{
```

```
Permission::create([
```

```
'group_id' => $group,
```

```
'name' => $permissions[$i]
```

```
]);
```

```
}
```

```
}
```

```
}
```

```
DB::commit();
```

```
Session::flash('success','Permission saved
successfully.');
```

```
} catch (\Exception $e) {
```

```
DB::rollback();
```

```
echo "something
```

```
wrong";die();
```

```
}
```

```
return redirect()-
```

```
>route('setting.group_permissions');
```

```
}
```

- Function Create Group Permission

```
public function createGroupPermission(){
```

```
return
```

```
view('backend.permissions.group_create');
```

```
}
```

```
public function
```

```
saveGroupPermission(Request $request){
```

```
$this->validate($request, [
```

```
'name' =>
```

```
'required|max:255|unique:group_permission
s,name',
```

```

]);

$group = GroupPermission::create([
    'name' => $request->name,
]);

if($group){
    Session::flash("message",
__('label.add_success'));

    Session::flash("alert-class", 'bg-
success');
}else{
    Session::flash("message",
__('label.add_fail'));

    Session::flash("alert-class", 'bg-
danger');
}

return redirect()-
>route('setting.create_group_permission');
}

```

- Function Edit Group Permission

```

public function
editGroupPermission(GroupPermission
$group){
    return
view('backend.permissions.group_edit',
compact('group'));
}

```

```

}

public function
updateGroupPermission(GroupPermission
$group, Request $request){
    $this->validate($request, [
        'name' =>
'required|max:255|unique:group_permission
s,name,',$group->id.',id',
    ]);

    $group->name = $request-
>name;

    $group->save();

    if($group){
        Session::flash("message",
__('label.updated_success'));

        Session::flash("alert-class", 'bg-
success');
    }else{
        Session::flash("message",
__('label.updated_fail'));

        Session::flash("alert-class", 'bg-
danger');
    }

    return redirect()-
>route('setting.group_permissions');
}

```

❖ Middleware

➤ Group Permission

```
public function handle($request, Closure
$next)
{
    $route = Route::currentRouteName();

    $group = Auth::user()->group_id;

    $permission =
Permission::where('name',$route)-
>where('group_id',$group)->first();

    if(!$permission && $group!=1){

        Session::flash("message",
__("Sorry, You don't have any permission to
perform the action. Please contact
administrator for help."));

        Session::flash("alert-class", 'bg-
danger');

        return redirect()->route('home');

    }

    return $next($request);
}
```

❖ Script JS

➤ Reservation and Check in

- Script for form create and edit

```
$(document).ready(function(){

    var discountType =
$('#totalDiscountType');
```

```
if(discountType.val()=='discount_per
cent'){

    $('#btnTotalDiscountChange').html('
%');

}else{

    $('#btnTotalDiscountChange').html('<
i class="fa fa-money"></i>');

}

});

$(document).on('click', '#changeDisType',
function(){

    var discountType =
$('#discountType');

    if(discountType.val()=='discount_per
cent'){

        discountType.val('discount_amount');

        $(this).html('<i class="fa fa-
money"></i>');

    }else{

        discountType.val('discount_percent');

        $(this).html('%');

    }

    $('#discount').val(0);
```

<pre> }); \$(document).on('keyup', '#discount', function(){ var discountType = \$('#discountType'); var amount = parseFloat(\$('#amount').val()); var eVal = \$(this).val(); if(!amount amount<0){ swal('Amount Required!',", 'warning'); \$(this).val(0); return false; } if(discountType.val()=='discount_per cent'){ if(\$(this).val(>100){ swal("Biggest 100%", "", "error"); \$(this).val(0); } }else{ if(eVal>amount){ swal("Please Check Amount!", "", "error"); </pre>	<pre> \$(this).val(0); return false; } } }); \$(document).on('click', 'body #btnTotalDiscountChange', function(){ var discountType = \$('#totalDiscountType'); if(discountType.val()=='discount_per cent'){ discountType.val('discount_amount'); \$(this).html('<i class="fa fa- money"></i>'); }else{ discountType.val('discount_percent'); \$(this).html('%'); } \$('#totalDiscount').val(0); loadItem(); }); \$(document).on('change', '#totalDiscount', function){ </pre>
---	--

```

        var discountType =
$('#totalDiscountType');

        if(discountType.val()=='discount_per
cent'){

            if($(this).val()>100){

                swal("Biggest
100%", "", "error");

                $(this).val(0);

            }

        }else{

            if($(this).val()>$('#grandTotal').find('i
nput').val()*1){

                swal("Please Check
Grand Total", "", "error");

                $(this).val(0);

            }

        }

        loadItem();

    })

$(document).on('change', '#staying_type',
function(){

    var guest = $('#guest').val();

    if(guest==0){

```

```

        swal("Guest
Required!", "", "warning");

        $('#staying_type').val(0);

        return false;

    }

})

$(document).on('click', '#btnReset',
function(){

    swal({

        title:"Reset",

        text:"Are you sure you want
to reset?",

        icon:"warning",

        buttons:{

            cancel:{

                text:"No",value:null,visible:!0,className:
"btn-danger",closeModal:1

            },

            confirm:{

                text:"Yes",value:!0,visible:!0,className:
"btn-primary",closeModal:1

            }

        }

    }).then((confirm) => {

        if (confirm) {

```

```

        localStorage.removeItem('rlItem');

        location.reload();

    }

});

function addlItem(){

    var room = $('#room
option:selected').val();

    var room_name = $('#room
option:selected').text();

    var room_type =
$('#room_type').val();

    var room_type_name =
$('#room_type option:selected').text();

    var staying_type =
$('#staying_type').val();

    var staying_type_name =
$('#staying_type option:selected').text();

    var duration = $('#duration').val();

    var date_from = $('#dateFrom').val();

    var date_to = $('#dateTo').val();

    var amount = $('#amount').val();

    var discount = $('#discount').val();

    var discount_type =
$('#discountType').val();

```

```

    if(!room){

        swal('Please Select
Room!', '', 'warning')

        $('#btnAddlItem')[0].disabled
= true;

        return false;

    }

    if(!discount){

        discount=0;

    }

    if(!amount || amount<0){

        swal('Amount Required!', '',
'warning');

        return false;

    }

    var rowlItem={

        room_no:room,

        room_name:room_name,

        room_type:room_type,

        room_type_name:room_type_name,

        staying_type:staying_type,

        staying_type_name:staying_type_na
me,

        duration:duration,

```

<pre> date_from:date_from, date_to:date_to, amount:amount, discount:discount, discount_type:discount_type }; var items = JSON.parse(localStorage.getItem('rItem')); var dup=0; var dupIndex; if(items){ \$.each(items, function(index, item){ if(item.room_no==room){ dup=1; dupIndex = index; } }); if(dup){ items[dupIndex]=rowItem; }else{ items.push(rowItem) </pre>	<pre> } }else{ items=[]; items.push(rowItem); } localStorage.setItem('rItem', JSON.stringify(items)); loadItem(); \$('#btnAddItem')[0].disabled = true; }; \$(document).on('click', 'body .cancel-row- item', function(){ swal({ title:"Remove Item", text:"Are you sure you want to remove this item?", icon:"warning", buttons:{ cancel:{ text:"No",value:null,visible:!0,className:"btn-danger",closeModal:1 }, </pre>
---	--

```

        confirm:{
            text:"Yes",value:!0,visible:!0,className:
me:"btn-primary",closeModal:1
        }
    }
}).then((confirm) => {
    if (confirm) {
        var eThis = $(this);
        var parents =
eThis.parents('tr');
        var iIndex =
parents.find('.row_item_no').val();
        var items =
JSON.parse(localStorage.getItem('rlItem'));
        items.splice(iIndex,1);
        localStorage.setItem('rlItem',
JSON.stringify(items));
        loadItem();
    }
});

$(document).on('change', '#deposit',
function(){
    var grandTotal =
$('#grandTotal').find('input').val();

```

```

grandTotal = parseFloat(grandTotal);
e = parseFloat($(this).val());
if(e>grandTotal){
    swal("Warning","Please
Check Grand Total!","warning");
    $(this).val(0);
}
loadItem();
});

function loadItem(){
    var items =
JSON.parse(localStorage.getItem("rlItem"));
    var rowHTML = "";
    var sub_total =0;
    var totalDiscountType =
$('#totalDiscountType').val();
    var totalDiscount =
$('#totalDiscount').val();
    var tDiscount =0;
    var grand_total =0;
    var deposit = $('#deposit').val();
    if(!deposit){
        deposit=0;
    }

```

```

        if(!totalDiscount){
            totalDiscount=0;
        }

        $.each(items, function(index,item){

            var total_amount =
item.amount * item.duration;

            var itemdiscount = 0;

            if(item.discount_type=='discount_per
cent'){

                itemdiscount =
total_amount*(item.discount/100);

            }else{

                itemdiscount =
item.discount;

            }

            total_amount = total_amount
- itemdiscount;

            sub_total += total_amount;

            rowHTML +=<tr>';

            rowHTML +=<td class="text-
right"><input type="hidden"
class="row_item_no"
value="" +index+"">' + (index+1) + '</td>';

            rowHTML +=<td><input
type="hidden" name="room_ids[]"

```

```

value="" +item.room_no+"">' +item.room_nam
e+ '</td>';

            rowHTML +=<td><input
type="hidden" name="room_type_ids[]"
value="" +item.room_type+"">' +item.room_typ
e_name+ '</td>';

            rowHTML +=<td><input
type="hidden" name="staying_type_ids[]"
value="" +item.staying_type+"">' +item.staying
_type_name+ '</td>';

            rowHTML +=<td class="text-
right"><input type="hidden"
name="durations[]"
value="" +item.duration+"">' +item.duration+ '</
td>';

            rowHTML +=<td><input
type="hidden" name="checkins[]"
value="" +item.date_from+"">' +item.date_fro
m+ '</td>';

            rowHTML +=<td><input
type="hidden" name="checkouts[]"
value="" +item.date_to+"">' +item.date_to+ '</t
d>';

            rowHTML +=<td class="text-
right"><input type="hidden"
name="amounts[]"
value="" +item.amount+"">' +item.amount+ '</
td>';

```

```

        if(item.discount_type=="discount_per
cent"){

            rowHTML +=<td
class="text-right"><input type="hidden"
name="discounts[]"
value="" +item.discount+"><input
type="hidden" name="discount_types[]"
value="" +item.discount_type+">' +item.disco
unt+'%</td>';

        }else{

            rowHTML +=<td
class="text-right"><input type="hidden"
name="discounts[]"
value="" +item.discount+"><input
type="hidden" name="discount_types[]"
value="" +item.discount_type+">${item.disc
ount+'</td>';

        }

        rowHTML +=<td class="text-
right">${total_amount+'</td>';

        rowHTML +=<td>';

        rowHTML +=<a class="text-
danger cancel-row-item"><i class="fa fa-
trash-o"></i> remove</a>';

        rowHTML +=</td>';

        rowHTML +=</tr>';

    });

    $('#bodyItem').html(rowHTML);

```

```

    $('#subTotal').find('input').val(sub_tot
al);

    $('#subTotal').find('span').text(sub_to
tal);

    if(totalDiscountType=='discount_perc
ent'){

        tDiscount =
sub_total*(totalDiscount/100);

    }else{

        tDiscount = totalDiscount;

    }

    grand_total = sub_total -
(1.0*tDiscount+deposit*1.0);

    $('#grandTotal').find('input').val(gran
d_total);

    $('#grandTotal').find('span').text(gran
d_total);

}

```

❖ View

➤ Form Create Reservation and Check in

```

@extends('backend.layouts.app')

@section('title', $title.' |
'.$AppProfile['name'])

@section('content')

    <link rel="stylesheet" type="text/css"
href="{{asset('backend/app-

```

```

assets/vendors/css/pickers/daterange/dater
angepicker.css'}}}">

<link rel="stylesheet" type="text/css"
href="{{ asset('backend/app-
assets/vendors/css/pickers/datetime/bootstr
ap-datetimepicker.css') }}">

<link rel="stylesheet" type="text/css"
href="{{asset('backend/app-
assets/vendors/css/pickers/pickadate/pickad
ate.css')}}">

<style type="text/css">

    #changeDisType{

        cursor: pointer;

    }

    .table tr td, .table tr th{

        padding: 6px
8px!important;

    }

</style>

<!-- BEGIN: Content-->

<div class="app-content content">

    <div class="content-wrapper">

        <div class="content-header row">

            <div class="content-header-left col-
md-6 col-12 mb-2">

                <h3 class="content-header-title mb-
0">{{ __('label.rooms') }}</h3>

```

```

<div class="row breadcrumbs-top">

    <div class="breadcrumb-wrapper
col-12">

        <ol class="breadcrumb">

            <li class="breadcrumb-item"><a
href="{{ route('home') }}">{{ __('title.home')
}}</a>

            </li>

            <li class="breadcrumb-item
active"><a href="#">{{ $title_add }}</a>

            </li>

        </ol>

    </div>

</div>

</div>

</div>

<div class="content-body"><!-- Basic
form layout section start -->

<section id="horizontal-form-layouts">

    <div class="row">

        <div class="col-md-12">

            @if(Session::has('message'))

                <div class="alert {{
Session::get('alert-class') }} alert-icon-left
alert-dismissible mb-2" role="alert">

```

```

                                <button
type="button" class="close" data-
dismiss="alert" aria-label="Close">

                                <span
aria-hidden="true">×</span>

                                </button>

                                <strong>{{
Session::get('message') }}</strong>

                                </div>

                                @endif

                                <div class="card">

                                    <div class="card-header">

                                        <h4 class="card-title"
id="horz-layout-basic">{{ $title_add }}</h4>

                                        <a class="heading-
elements-toggle"><i class="fa fa-ellipsis-v
font-medium-3"></i></a>

                                        <div class="heading-
elements">

                                            <ul class="list-inline
mb-0">

                                                <li><a data-
action="collapse"><i class="ft-
minus"></i></a></li>

                                                <li><a data-
action="reload"><i class="ft-rotate-
cw"></i></a></li>

```

```

                                                <li><a data-
action="expand"><i class="ft-
maximize"></i></a></li>

                                                {{-- <li><a data-
action="close"><i class="ft-x"></i></a></li> -
-}}

                                                </ul>

                                                </div>

                                                </div>

                                                <div class="card-content
collapse show">

                                                    <div class="card-body">

                                                        @if($checkin==1)

                                                            {{
Form::open(['route'=>
['checkin.create','checkin'=>1],'class' =>
'form form-horizontal', 'method' => 'POST',

                                                            'enctype' =>
'multipart/form-data']) }}

                                                        @else

                                                            {{
Form::open(['route'=>
'reservation.create','class' => 'form form-
horizontal', 'method' => 'POST',

                                                            'enctype' =>
'multipart/form-data']) }}

                                                        @endif

```

```

        <div class="form-
body">

        <div class="row">

            <div class="col-md-4
col-sm-6">

                <div
class="form-group {{ ($errors-
>has('guest'))?'error':" }}">

                    <label
class="label-control"
for="guest">Guest</label>

                    <div
class="position-relative has-icon-left">

                        {!! Form::select('guest', $guests,null,
['class' => 'select2-data-array form-control',
'style' => 'width:100%!important;', 'id' =>
'guest', 'required'=>'true']) !!}

                        @if($errors->has('guest'))

                            <div class="help-block"><ul

```

```

role="alert"><li>{{ $errors->first('guest')
}}</li></ul></div>

                                @endif

                                </div>

                                </div>

                                </div>

                                <div class="col-md-4
col-sm-6">

                                    <div
class="form-group {{ ($errors-
>has('staying_type'))?'error':" }}">

                                        <label
class="label-control"
for="staying_type">Staying Type</label>

                                        <div
class="position-relative has-icon-left">

                                            {!! Form::select('staying_type',
$staying_types,null, ['class' => 'select2-data-
array form-control', 'style' =>
'width:100%!important;', 'id' =>
'staying_type']) !!}

```

```

</div>

</div>

</div>

<div class="col-md-4
col-sm-6">

    <div
class="form-group {{ ($errors-
>has('duration'))?'error':" }}">

        <label
class="label-control"
for="duration">Duration</label>

        <div
class="position-relative has-icon-left">

            {!! Form::number('duration', null,
['class' => 'form-control',

            'placeholder' => 'duration', 'id' =>
'duration', 'min'=>'1', 'oninput'=>"this.value =
this.value.replace(/[^\0-9]/g,
'').replace(/(\.\.*)" /g, '$1');") !!}

```

```

<div class="form-control-position">

    <i class="ft-clock"></i>

</div>

</div>

@if($errors->has('duration'))

    <div class="help-block"><ul
role="alert"><li>{{ $errors->first('duration')
}}</li></ul></div>

    @endif

</div>

</div>

<div class="col-md-4
col-sm-6">

    <div
class="form-group">

        <label

```

```
class="label-control" for="date">Checkin  
Date</label>
```

```
<div  
class="input-group">
```

```
<div class="input-group-prepend">
```

```
<span class="input-group-text">
```

```
<span class="fa fa-calendar-  
o"></span>  
</span>
```

```
</div>
```

```
{!! Form::date('date', null, ['class' =>  
'form-control pickadate date-frm', 'id' =>  
'dateFrom']) !!}
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="col-md-4  
col-sm-6">
```

```
<div  
class="form-group">
```

```
<label  
class="label-control" for="date">Checkout  
Date</label>
```

```
<div  
class="input-group">
```

```
<div class="input-group-prepend">
```

```
<span class="input-group-text">
```

```
<span class="fa fa-calendar-  
o"></span>  
</span>  
</div>
```

```
{!! Form::text('date', null, ['class' =>  
'form-control pickadate-arrow date-frm',  
'disabled', 'id' => 'dateTo']) !!}
```

```

</div>

</div>

</div>

<div class="col-md-4
col-sm-6">

    <div
class="form-group {{ ($errors-
>has('room_type'))?'error':" }}">

        <label
class="label-control" for="room_type">Room
Type</label>

        <div
class="position-relative has-icon-left">

            {!!
Form::select('room_type', $room_types,null,
['class' => 'select2-data-array form-control',
'style' => 'width:100%!important;', 'id' =>
'room_type']) !!}

        </div>
    </div>
</div>

</div>

```

```

<div class="col-md-4
col-sm-6">

    <div class="form-
group {{ ($errors->has('room'))?'error':"
}}">

        <label
class="label-control"
for="room">Room</label>

        <div
class="position-relative has-icon-left"
id="boxRoom">

            {!! Form::select('room', $rooms,null,
['class' => 'select2-data-array form-control',
'style' => 'width:100%!important;', 'id' =>
'room']) !!}

        </div>
    </div>

</div>

<div class="col-md-4
col-sm-6">

    <div class="form-group {{
($errors->has('amount'))?'error':" }}">

        <label

```

```

class="label-control"
for="amount">Amount</label>

<div
class="position-relative has-icon-left">

    {!! Form::number('amount', null,
    ['class' => 'form-control',

        'placeholder' => 'amount', 'id' =>
        'amount', 'min'=>'0', 'oninput'=>"this.value =
        this.value.replace(/^[^0-9.]/g,
        '').replace(/(\.\.*)" /g, '$1');") !!}

        <div class="form-control-position">

            <i class="fa fa-money"></i>
        </div>

        </div>

        @if($errors-
        >has('amount'))

            <div class="help-
            block"><ul role="alert"><li>{{ $errors-
            >first('amount') }}</li></ul></div>

            @endif

        </div>

        </div>

        <div class="col-md-4 col-sm-
6">

```

```

<div
class="form-group {{ ($errors-
>has('discount'))?'error':' }}">

    <label

class="label-control"
for="discount">Discount</label>

        {{-- {!!
Form::hidden('adsfsf', 'discount_percent',
['id'=>'discountType']) !!} --}}

        <input
type="hidden" id="discountType"
value="discount_percent">

        <div class="position-
relative has-icon-left">

            {!! Form::number('discount', null,
            ['class' => 'form-control',

                'placeholder'
=> 'discount', 'id' => 'discount', 'min'=>'0',
'oninput'=>"this.value =
this.value.replace(/^[^0-9.]/g,
'').replace(/(\.\.*)" /g, '$1');") !!}

            <div class="form-control-position"
id="changeDisType">

```

```

%
</div>
</div> @if($errors-
>has('discount')) <div
class="help-block"><ul role="alert"><li>{{
$errors->first('discount') }}</li></ul></div>

@endif

</div>

</div>

<div class="col-md-12
mt-2"> <button
type="button" class="btn btn-save btn-
primary pull-right" id="btnAddItem">

<i
class="fa fa-plus"></i> Add
</button>

</div>

@if($errors->any())

<div class="col-md-12
mb-1">
<h6 class="text-danger" style="text-

```

```

decoration: underline;">Reservation
Validation Require</h6>

```

```

<ul>

@foreach($errors->all() as $error)

<span class="help-block text-
danger">

<li><strong>{{ $error
}}</strong></li>

</span>

@endforeach

</ul>

</div>

@endif

<div class="col-md-12
mt-2 mb-2">
<div class="table-
responsive">

<table
class="table table-bordered table-nowrap">

```

```

<thead>
<tr>
<th>No</th>
<th>Room</th>
<th>Room Type</th>
<th>Staying Type</th>
<th>Duration</th>
<th>Checkin</th>
<th>Checkout</th>
<th>Amount</th>
<th>Discount</th>
<th style="min-width:
150px;">Total</th>

<th>Function</th>

</tr>

</thead>

<tbody id="bodyItem">

</tbody>

```

```

<tfoot>

<tr>
<th colspan="9" class="text-right">Sub
Total</th>
<th class="text-right" id="subTotal">
<input
type="hidden" name="sub_total" value="0">
$<span></span>
</th>

<th></th>
</tr>
<tr>
<th colspan="9" class="text-right">
Discount
<button type="button"
class="btn btn-sm btn-outline-primary"
id="btnTotalDiscountChange">%</button>
</th>
<th class="text-right">
{{-- <input
name="total_dicount_type" type="hidden"
id="totalDiscountType"
value="discount_percent"> --}}

```

```

{{-- <input
name="total_dicount" type="number"
min="0" width="20px" id="totalDiscount"
value="0"> --}}

```

```

{!!
Form::hidden('total_discount_type',
'discount_percent', ['id'=>
'totalDiscountType']) !!}

```

```

{!!
Form::number('total_discount', 0, ['id'=>
'totalDiscount' , 'min' => 0,
'oninput'=>"this.value =
this.value.replace(/^[^0-9.]/g,
'').replace(/(\.)*\./g, '$1');"]) !!}
</th>

```

```

<th></th>
</tr>

```

```

<tr @if($checkin==1) style="display:
none;" @endif>

```

```

<th colspan="9" class="text-
right">Deposit</th>
<th class="text-right">

```

```

{{-- <input
name="deposit" type="number" min="0"
width="20px" id="deposit" value="0"> --}}

```

```

{!!
Form::number('deposit', 0, ['id'=> 'deposit',
'min' => 0, 'oninput' =>"this.value =
this.value.replace(/^[^0-9.]/g,
'').replace(/(\.)*\./g, '$1');"]) !!}

```

```

</th>

```

```

<th></th>

```

```

</tr>

```

```

<tr>

```

```

<th colspan="9"

```

```

class="text-right">Grand Total</th>

```

```

<th class="text-right"

```

```

id="grandTotal">

```

```

<input type="hidden"

```

```

name="grand_total" value="0">

```

```

$<span></span>

```

```

</th>

```

```

<th></th>

```

```

</tr>

```

```

</tfoot>

```

```

</table>

```

```

</div>

```

```

        </div>

        </div>

        {{-- table --}}

        </div>

        {{-- row --}}

    </div>

    <div class="form-actions"
align="right">
        <button type="reset" class="btn btn-
cancel btn-danger mr-1" id="btnReset">

                <i class="ft-
x"></i> {{ __( 'label.cancel' ) }}
            </button>
        <button type="submit"
class="btn btn-save btn-primary">

                <i
class="fa fa-check-square-o"></i> {{
__( 'label.save' ) }}
            </button>
    </div>
    </div>

    {{Form::close()}}

    </div>

    </div>

</div>

```

```

        </div>

    </div>

</section>

<!-- // Basic form layout section end -->

    </div>

    </div>

    </div>

    <!-- END: Content-->

@endsection

@push('scripts')

    <script
src="{{asset('backend/reservation/script.js')
}}"></script>

    <script src="{{asset('backend/app-
assets/vendors/js/pickers/dateTime/bootstra
p-datetimepicker.min.js')}}"></script>

    <script src="{{asset('backend/app-
assets/vendors/js/pickers/pickadate/picker.js
')}}"></script>

    <script src="{{asset('backend/app-
assets/vendors/js/pickers/pickadate/picker.d
ate.js')}}"></script>

    <script src="{{asset('backend/app-
assets/vendors/js/pickers/pickadate/picker.ti
me.js')}}"></script>

```

```

<script src="{asset('backend/app-
assets/vendors/js/pickers/pickadate/legacy.j
s')}}"></script>

<script src="{asset('backend/app-
assets/vendors/js/pickers/daterange/dateran
gepicker.js')}}"></script>

<script src="{asset('backend/app-
assets/js/scripts/ui/breadcrumbs-with-
stats.min.js')}}"></script>

<script src="{asset('backend/app-
assets/js/scripts/pickers/dateTime/pick-a-
datetime.min.js')}}"></script>

<script type="text/javascript">

    $(document).ready(function(){

        $('#btnAddItem')[0].disabled
= true;

        loadItem();

    })

    $(document).on('click','#btnAddItem',
function(){

        addItem();

    })

    $(document).on('change',
'#dateFrom', function(){

        var datefrom = $(this).val();

        var duration =
$('#duration').val();

```

```

        var staying =
$('#staying_type').val();

        $('#boxRoom').html('{!!
Form::select('room', $rooms,null, ['class' =>
'select2-data-array form-control', 'style' =>
'width:100%!important;', 'id' => 'room']) !!}');

        $('#amount').val("");

        $('#discount').val("");

        $('#btnAddItem')[0].disabled
= true;

        if(staying==0){

            swal("Staying Type
Required!", "", "warning");

            $(this).val("");

            return false;

        }

        if(!duration){

            swal("Duration
Required!", "", "warning");

            $(this).val("");

            $( "#duration"
).focus();

        }

        if(!$ (this).val()){

            return false;

        }

```

```

        $.ajax({
            type:'get',
            url:'{{
route('get_dateto') }}',

            data:{datefrom:datefrom,
duration:duration},

            success:function(data){

                $('#dateTo').val(data);

            }

        });

        $(document).on('change',
'#room_type', function(){

            var date_from =
$('#dateFrom').val();

            var duration =
$('#duration').val();

            if(!date_from){

                swal('Checkin
Required!', '', 'warning');

                $('#room_type').val(0);

                return false;

            }

```

```

        var room_type =
$(this).val();

        $.ajax({

            type:'get',

            url:'{{
route('get_rooms') }}',

            data:{room_type_id:room_type,
date_from:date_from, duration:duration},

            success:function(data){

                $('#boxRoom').html(data);

                $('#amount').val(0);

                $('#btnAddItem')[0].disabled = true;
                $('#room').select2();

            }

        });

        $(document).on('change','body
#room', function(){

            var room =
$(this).val();

            var staying_type =
$('#staying_type').val();

```

```

        if($(this)[0].selectedIndex==0){
            $('#btnAddItem')[0].disabled = true;

            return false;

        }

        $.ajax({

            type:'get',

            url:'{{

route('get_room_price') }}',

            data:{room_id:room,
staying_type:staying_type},

            success:function(data){

                $('#amount').val(data);

            }

        });

        $('#btnAddItem')[0].disabled = false;

    });

</script>

@endpush

```