

សាកលវិទ្យាល័យភូមិន្ទនីតិសាស្ត្រនិងវិទ្យាសាស្ត្រសេដ្ឋកិច្ច

Université Royale de Droit et des Sciences Économiques
Royal university of Law and Economics



សារណាបញ្ចប់ការសិក្សា

ការបេសាគេហទំព័រ

ស្រាវជ្រាវពីថ្ងៃទី ០៣ ខែមេសា ឆ្នាំ ២០២៣ ដល់ថ្ងៃទី ៣១ ខែឧសភា ឆ្នាំ ២០២៣

ស្រាវជ្រាវដោយ

និស្សិតឈ្មោះ ៖ លោក **អ៊ុន ម៉ានិត**

លោក **ម៉ៅ សុខណាត**

ណែនាំដោយ

សាស្ត្រាចារ្យជំនួយបណ្ឌិត **ទិន ហេង**

ថ្នាក់បរិញ្ញាបត្រ **សេដ្ឋកិច្ចព័ត៌មានវិទ្យា**

ជំនាន់ទី ១៨

ឆ្នាំចូលសិក្សា

២០១៩

ឆ្នាំសរសេរសារណា

២០២៣

សេចក្តីផ្តើមអំណាច

ក្នុងនាមពួកយើងជានិស្សិតបរិញ្ញាបត្រសេដ្ឋកិច្ចព័ត៌មានវិទ្យាជំនាន់ទី ១៨ ឆ្នាំសិក្សា ២០១៩ - ២០២០ នៃសាកលវិទ្យាល័យភូមិន្ទនីតិសាស្ត្រ និងវិទ្យាសាស្ត្រសេដ្ឋកិច្ច ពួកយើងសូមធ្វើការគោរពដឹងគុណ និងថ្លែងអំណរគុណយ៉ាងជ្រៀលជ្រៅចំពោះ៖

រាជរដ្ឋាភិបាល និងថ្នាក់ដឹកនាំនៃប្រទេស ដែលបានផ្តល់នូវសុខសន្តិភាព ស្ថេរភាព ហេដ្ឋារចនាសម្ព័ន្ធដូចជា មន្ទីរពេទ្យ ផ្លូវថ្នល់ និងជាពិសេសនោះគឺសាលារៀនដែលជាកន្លែងផ្តល់ការអប់រំ និងបានផ្តល់ឱកាសឱ្យពួកយើងបានក្រេបនូវចំណេះដឹង និងវិជ្ជាជីវៈ។

លោកអ្នកមានគុណទាំងពីរ គឺលោកឪពុកអ្នកម្តាយដែលបានផ្តល់កំណើតឱ្យកូនបានឃើញពន្លឺថ្ងៃ មិនតែប៉ុណ្ណោះលោកទាំងពីរបានចិញ្ចឹមបីបាច់ថែរក្សា និងប្រៀនប្រដៅកូនឱ្យក្លាយជាមនុស្សល្អ។ លើសពីនេះទៅទៀតលោកទាំងពីរបានខិតខំតស៊ូជំនះរាល់ឧបសគ្គដើម្បីឱ្យកូនបានរៀនខ្ពង់ខ្ពស់មានចំណេះដឹងមានតម្លៃក្នុងសង្គម និងអនាគតល្អរុងរឿង។

ឯកឧត្តមសាកលវិទ្យាធិការ សាកលវិទ្យាធិការរង លោក និងលោកស្រីសាស្ត្រាចារ្យ ព្រមទាំងបុគ្គលិកនៃសាកលវិទ្យាល័យភូមិន្ទនីតិសាស្ត្រ និងវិទ្យាសាស្ត្រសេដ្ឋកិច្ចទាំងអស់ដែលបានផ្តល់ឱកាស និងអនុញ្ញាតឱ្យយើងខ្ញុំបានចូលសិក្សាតាំងពីឆ្នាំសិក្សាមូលដ្ឋានរហូតដល់បញ្ចប់ថ្នាក់បរិញ្ញាបត្រនៃជំនាញសេដ្ឋកិច្ចព័ត៌មានវិទ្យាជាស្ថាពរ។

សាស្ត្រាចារ្យជំនួយ បណ្ឌិត **ទិន ហេង** ដែលបានណែនាំ ក្នុងការសរសេរសារណាបញ្ចប់ការសិក្សាលោកបានផ្តល់នូវអនុសាស្ត្រ និងបណ្តុះគំនិតយោបល់ល្អៗជាច្រើន ព្រមទាំងចំណាយទាំងកម្លាំងកាយ កម្លាំងចិត្ត និងពេលវេលាយ៉ាងមានតម្លៃក្នុងការជួយកែសម្រួលពាក្យពេជ្រ និងអក្ខរាវិរុទ្ធក្នុងការសរសេរបញ្ចប់សារណានេះផងដែរ។ ដោយទទួលបាននូវការគាំទ្រលើកទឹកចិត្តពីសំណាក់លោកសាស្ត្រាចារ្យជាហេតុធ្វើឱ្យកិច្ចការស្រាវជ្រាវនេះបញ្ចប់ទៅបានយ៉ាងរលូន។

បងប្អូន មិត្តភក្តិ មិត្តរួមថ្នាក់ មិត្តរួមជំនាន់ និងសិស្សច្បងទាំងអស់ដែលបានជួយផ្តល់ជាបទពិសោធន៍លើកទឹកចិត្ត និងជម្រុញឱ្យយើងខ្ញុំបានបញ្ចប់កិច្ចការស្រាវជ្រាវនេះជាស្ថាពរ។

ជាចុងក្រោយ យើងខ្ញុំសូមប្រសិទ្ធិពរជ័យដល់រាជរដ្ឋាភិបាល ថ្នាក់ដឹកនាំ លោកឪពុក អ្នកម្តាយ ឯកឧត្តមសាកលវិទ្យាធិការ សាកលវិទ្យាធិការរង លោក លោកស្រីសាស្ត្រាចារ្យ លោកសាស្ត្រាចារ្យ បណ្ឌិត **ទិន ហេង** បងប្អូន មិត្តភក្តិ មិត្តរួមថ្នាក់ មិត្តរួមជំនាន់ សិស្សច្បងទាំងអស់គ្នា ព្រមទាំងបុគ្គលិកសាលាទាំងអស់ ទទួលបាននូវពុទ្ធពរជ័យទាំងបួនប្រការគឺ អាយុ វណ្ណៈ សុខៈ ពលៈ កុំបីឃ្លៀងឃ្លាតឡើយ។

អារម្ភកថា

ងាកទៅកំឡុងពេលដែលមេរោគ Covid-19 បានកំពុងរីករាលដាលពាសពេញសាកលលោក មនុស្សម្នាមានផលពិបាកជាខ្លាំងក្នុងការទិញទំនិញ ម្ហូបអាហារ សេវាកម្មផ្សេងៗ និងគ្រឿងប្រើប្រាស់ចាំបាច់ប្រចាំថ្ងៃដែលជាពេលវេលាមួយធ្វើឱ្យយើងនិយម និងសម្រុកទៅរកមធ្យោបាយទិញទំនិញតាមរយៈអនឡាញ។ ដោយសារតែការទិញទំនិញ និងសេវាកម្មតាមអនឡាញមានភាពងាយស្រួល និងដើម្បីបង្ការពីមេរោគមួយនោះផងដែរ។ ប៉ុន្តែក្នុងនោះក៏មានផលវិបាកខ្លះដែរ ដោយសារតែគេហទំព័រ ឬ Website ខ្លះពិបាកក្នុងការប្រើប្រាស់ វាមានភាពស្មុគស្មាញច្រើនធ្វើឱ្យអ្នកប្រើប្រាស់ពិបាកយល់។

ដោយហេតុបានមើលឃើញ និងបានដឹងពីបញ្ហាចម្បងខាងលើ ធ្វើឱ្យយើងខ្ញុំបានមានគំនិតរួមមួយលើការជ្រើសរើសប្រធានបទសម្រាប់កិច្ចការស្រាវជ្រាវបញ្ចប់ថ្នាក់បរិញ្ញាបត្រនេះដោយយើងនឹងលើកឡើងនូវប្រធានបទ៖ **ការរចនាគេហទំព័រ** ។ យើងខ្ញុំនឹងធ្វើការចងក្រងជាឯកសារ និងសៀវភៅសម្រាប់សិស្សប្អូន និងបងប្អូននិស្សិតដទៃទៀតដែលចង់ស្វែងយល់ពីការរចនាគេហទំព័រ។ នៅក្នុងតួសារណាទាំងមូលនឹងមានការលើកឡើងបង្ហាញពីការបង្កើតគេហទំព័រ និងរចនាគេហទំព័រមួយដែលមានភាពទាក់ទាញ និងមានភាពងាយស្រួលមិនស្មុគស្មាញដល់អ្នកប្រើប្រាស់។

ជាការពិតណាស់នៅខាងក្រោយសមិទ្ធផលមួយនេះ យើងខ្ញុំបានជួបផលវិបាកច្រើនរាប់មិនអស់ ប៉ុន្តែគ្រប់យ៉ាងប្រែជាធូរស្រាលប្រសើរឡើងវិញ ដោយមានការជ្រោមជ្រែង គាំទ្រ ដឹកនាំ សម្របសម្រួល និងណែនាំពីសំណាក់សាស្ត្រាចារ្យណែនាំជាទីគោរព និងការខិតខំប្រឹងប្រែងស្រាវជ្រាវ និងសិក្សាបន្ថែមរបស់យើងខ្ញុំ។ កត្តាទាំងអស់នេះហើយ បានធ្វើឱ្យកិច្ចការស្រាវជ្រាវមួយនេះបញ្ចប់ទៅដោយជោគជ័យជាស្ថាពរ។ ជាមួយគ្នានេះដែរ យើងខ្ញុំជឿថា គ្រប់យ៉ាងមិនអាចសុទ្ធតែល្អឥតខ្ចោះនោះទេ ហេតុដូច្នេះហើយយើងខ្ញុំអាចនឹងមានចំណុចខ្វះខាត ឬកំហុសឆ្គងណាមួយកើតមានជាក់ជាមិនខានឡើយ ដោយក្តីអនុគ្រោះសូមប្រិយមិត្តអ្នកអានទាំងអស់មេត្តាលើកលែង យោគយល់ និងអធ្យាស្រ័យដល់យើងខ្ញុំផង។

ជាចុងក្រោយយើងខ្ញុំសង្ឃឹមថាឯកសារ ឬសៀវភៅនេះអាចជួយជាផលប្រយោជន៍ដល់អ្នកអាន ក៏ដូចជាសិស្សប្អូនជំនាន់ក្រោយ។ យើងខ្ញុំនឹងបើកចិត្តទទួលយកការកែតម្រូវ និងវិវត្តន៍ក្នុងន័យស្ថាបនាជានិច្ច ដើម្បីការរីកចម្រើនទៅថ្ងៃមុខបន្តទៀត។ ទន្ទឹមនឹងនេះ យើងខ្ញុំសូមជូនពរដល់ប្រិយមិត្តអ្នកអានទាំងឡាយ ឱ្យជួបតែពុទ្ធពរទាំងបួនប្រការគឺ អាយុ វណ្ណៈ សុខៈ ពលៈ កុំបីឃ្លៀងឃ្លាតឡើយ។

មាតិកា

ទំព័រ

បញ្ជីអក្សរកាត់	IV
បញ្ជីដ្យាក្រាម.....	V
បញ្ជីតារាង	V
បញ្ជីរូបភាព.....	V

សេចក្តីផ្តើម

១. លំនាំបញ្ហា.....	១
២. ចំណោទបញ្ហា.....	២
៣. គោលបំណងនៃការស្រាវជ្រាវ.....	៣
៤. ដែនកំណត់ និងវិសាលភាពនៃការស្រាវជ្រាវ.....	៣
៥. សារៈសំខាន់នៃការស្រាវជ្រាវ.....	៣

ជំពូកទី ១

ទ្រឹស្តី

១. អ្វីទៅជាគេហទំព័រ?.....	៤
១.១ អ្វីទៅជា WWW ?	៤
១.២ អ្វីទៅជា Domain Name ?.....	៤
១.៣ អ្វីទៅជា Internet ?	៥
២. ការរចនាគេហទំព័រ (Web Design)	៦
២.១ ចំណុចប្រទាក់អ្នកប្រើប្រាស់ (UI = User Interface).....	៦
២.២ បទពិសោធន៍អ្នកប្រើប្រាស់ (UX = User Experience).....	៧
៣ ការរចនាក្រាហ្វិក (Graphic Design)	៧
៣.១ ប្លង់ (Layout).....	៨
៣.២ អក្សរ (Typography)	៨
៣.៣ ពណ៌ (Color).....	៩
៤ ការអភិវឌ្ឍន៍គេហទំព័រ (Web Development)	៩

៤.១ ការអភិវឌ្ឍន៍គេហទំព័រផ្នែកខាងមុខ (Front-end Development)	១០
៤.១.១ អ្វីទៅជា HTML ?.....	១១
៤.១.២ អ្វីទៅជា CSS ?	១២
៤.១.៣ អ្វីទៅជា Bootstrap ?	១២
៤.១.៣ អ្វីទៅជា Font-Awesome ?.....	១៣
៤.២ ការអភិវឌ្ឍន៍គេហទំព័រផ្នែកខាងក្រោយ (Back-end Development)	១៣
៤.២.១ អ្វីទៅជា JavaScript ?.....	១៤
៤.២.២ អ្វីទៅជា React JS ?	១៥
៤.២.៣ អ្វីទៅជា JQuery និង Ajax ?	១៥
៤.២.៤ អ្វីទៅជា PHP ?	១៦

ជំពូកទី ២

ស្ថានភាពទូទៅ

១. វិធីសាស្ត្រស្រាវជ្រាវ.....	១៨
១.១. ការជ្រើសរើសប្រធានបទស្រាវជ្រាវ.....	១៨
១.២. ការប្រមូលទិន្នន័យ.....	១៩
១.៣. ការកំណត់តម្រូវការព័ត៌មាន.....	១៩
១.៤. ការរៀបចំរចនាសម្ព័ន្ធនៃប្រព័ន្ធគេហទំព័រ	១៩
១.៥. ការសរសេរកូដ	២០
១.៦. ការធ្វើតេស្តសាកល្បង និងការថែទាំរក្សាគេហទំព័រ.....	២០
២. គម្រោងពេលវេលានៃការស្រាវជ្រាវ.....	២០

ជំពូកទី ៣

លទ្ធផលនៃការស្រាវជ្រាវ

១. ដំណើរការនៃការរចនាគេហទំព័រ.....	២១
២. ការរចនាលើផ្ទាំងវេបសាយ	២១
២.១. របៀប Login និង Register ទៅកាន់វេបសាយ.....	២១
២.២. ផ្ទៃខាងក្នុងនៃគេហទំព័រ.....	២៣

២.៣. ដំណើរការគ្រោងវេបសាយ	២៨
សេចក្តីសន្និដ្ឋាន	២៩
អនុសាសន៍	៣០
ឯកសារយោង	
ឧបសម្ព័ន្ធ	

បញ្ជីអក្សរកាត់

WWW = World Wide Web

UI = User Interface

UX = User Experience

http = Hypertext Transfer Protocol

HTML = Hypertext Markup Language

CSS = Cascading Style Sheets

QA = Quality Assurance

PHP = PHP: Hypertext Preprocessor

C# = C-Sharp

JS = JavaScript

Ajax = Asynchronous JavaScript and XML

XML = Extensible Markup Language

DOM = Document Object Model

DHTML = Dynamic Hypertext Markup language

API = Application Programming Interfaces

បញ្ជីជម្រក

ជម្រកទី ០១ ៖ បង្ហាញអំពីវីដេអូនៃតេហទំព័រ IMOVIE	២៩
--	----

បញ្ជីតារាង

តារាង ៖ បង្ហាញអំពីពេលវេលានៃកិច្ចស្រាវជ្រាវ	២១
--	----

បញ្ជីរូបភាព

រូបភាពទី ០១ ៖ បង្ហាញពីតេហទំព័រ MisterGatesDirect	០២
រូបភាពទី ០២ ៖ បង្ហាញពីតេហទំព័រ GateNFences.....	០២
រូបភាពទី ០៣ ៖ តេហទំព័រ freeCodeCamp.....	១១
រូបភាពទី ០៤ ៖ Logo នៃភាសា JavaScript HTML និង CSS	១២
រូបភាពទី ០៥ ៖ Logo របស់ Bootstrap	១៣
រូបភាពទី ០៦ ៖ Logo របស់ Font-Awesome.....	១៤
រូបភាពទី ០៧ ៖ Logo របស់ React JS	១៦
រូបភាពទី ០៨ ៖ Logo របស់ jQuery និង Ajax	១៧
រូបភាពទី ០៩ ៖ Logo របស់ PHP	១៨
រូបភាពទី ១០ ៖ បង្ហាញពីផ្ទាំងដំបូងតេហទំព័រ.....	២២
រូបភាពទី ១១ ៖ បង្ហាញពី Sign Up Form	២៣
រូបភាពទី ១២ ៖ បង្ហាញពី Sign In Form.....	២៣
រូបភាពទី ១៣ ៖ បង្ហាញពីផ្នែកខាងក្នុងនៃតេហទំព័រ	២៤
រូបភាពទី ១៤ ៖ បង្ហាញពី Slide នៃខ្សែភាពយន្តនៅលើ Home page	២៤
រូបភាពទី ១៥ ៖ បង្ហាញពីខ្សែភាពយន្តដែលកំពុងល្បី	២៥
រូបភាពទី ១៦ ៖ បង្ហាញពីខ្សែភាពយន្តដែលមានការ RATE ខ្ពស់.....	២៥
រូបភាពទី ១៧ ៖ បង្ហាញពីខ្សែភាពយន្តដែលនឹងចាក់ផ្សាយឆាប់ៗខាងមុខ	២៦
រូបភាពទី ១៨ ៖ បង្ហាញពីខ្សែភាពយន្តដែលកំពុងចាក់ផ្សាយ	២៦
រូបភាពទី ១៩ ៖ បង្ហាញពីប៊ូតុង Load more.....	២៧
រូបភាពទី ២០ ៖ បង្ហាញពីប៊ូតុង Search	២៧
រូបភាពទី ២១ ៖ បង្ហាញពីផ្នែកខាងក្នុងនៃខ្សែភាពយន្តនីមួយៗ	២៨
រូបភាពទី ២២ ៖ បង្ហាញពីផ្ទាំងសម្រាប់មើល trailer នៃខ្សែភាពយន្ត	២៨

សេចក្តីផ្តើម

១. លំដាប់ក្រោយ

បច្ចុប្បន្នពិភពលោកបានធ្វើការបោះជំហានទៅមុខជានិច្ច ហើយអ្វីៗទាំងអស់ដែលយើងឃើញ ឬធ្លាប់ឆ្លងកាត់បទពិសោធន៍គឺ សុទ្ធសឹងតែកើតឡើងដោយសារប្រព័ន្ធឌីជីថល (Digitalized)។ បកនិយាយទៅកាន់ ១៧ឆ្នាំមុន កាលពីឆ្នាំ ២០០៦ ទីផ្សារតាមអ៊ីនធឺណិតគឺ ធំរិសសវិសាល ប៉ុន្តែនៅតែមានរបស់របរមួយចំនួនតូចនៅតែតម្រូវឱ្យយើងទៅទិញដោយផ្ទាល់ ប៉ុន្តែសព្វថ្ងៃនេះអ្វីៗទាំងអស់ដែលពួកយើងចង់បានគ្រាន់តែទៅរកមើលនៅលើអ៊ីនធឺណិតរួចចុចទិញជាការស្រេច។ នេះគឺជាចំណុចដែលពិភពលោកត្រូវបានផ្លាស់ប្តូរ ហើយប្រសិនបើអ្នកមានអារម្មណ៍ថាតាមមិនទាន់អ្វីដែលពិភពលោកផ្លាស់ប្តូរទេនោះ ខ្ញុំគិតថាវាដល់ពេលដែលអ្នកត្រូវចាប់ផ្តើមដើរទៅមុខឱ្យទាន់នឹងនិន្នាការនៃការផ្លាស់ប្តូររបស់មនុស្សក្នុងយុគសម័យថ្មីហើយ។ ហេតុនេះហើយរឿងសំខាន់មួយដែលអ្នកជំនួញ ម្ចាស់អាជីវកម្ម ឬក្រុមហ៊ុនធំតូចត្រូវធ្វើនោះគឺ ការបង្កើត ឬអភិវឌ្ឍន៍គេហទំព័រមួយដ៏ល្អដើម្បីទាក់ទាញចិត្តអ្នកប្រើប្រាស់ឱ្យចូលមកទស្សនាក៏ដូចជាទិញផលិតផលយើង។ ជាក់ស្តែងយោងតាមគេហទំព័រ

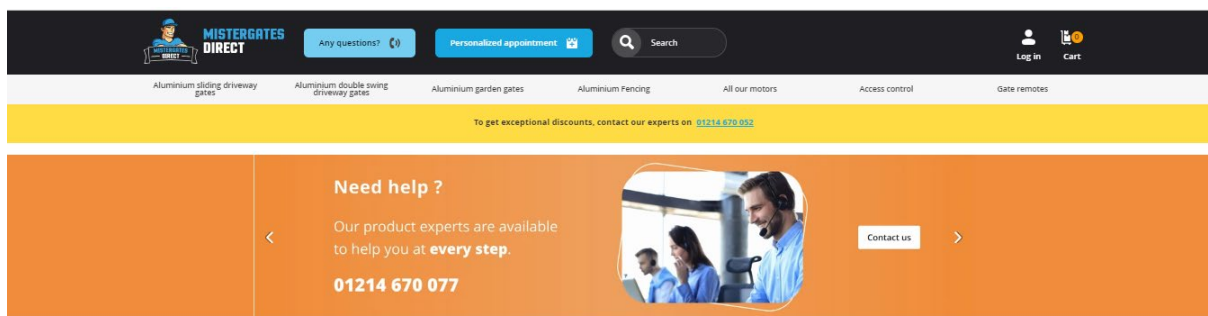
<https://www.statista.com/statistics/617136/digital-population-worldwide/> ដោយគិតត្រឹមខែមេសា

ឆ្នាំ ២០២៣ មានចំនួនអ្នកប្រើប្រាស់ព័ន្ធអ៊ីនធឺណិតរហូតទៅដល់ ៥.១៨ពាន់លាននាក់ ឬ ៦៤.៦% នៃចំនួនប្រជាជននៅលើពិភពលោក ហើយចំនួនប្រជាជនដែលមានបណ្តាញសង្គមប្រើប្រាស់មានចំនួន ៤.៨ពាន់លាននាក់ ឬ ៥៩.៩% នៃចំនួនប្រជាជនសរុបនៅលើពិភពលោក។ សព្វថ្ងៃនេះមនុស្សអាចធ្វើអ្វីៗបានទាំងអស់នៅលើគេហទំព័រសូម្បីតែការជជែកជាមួយមិត្តភក្តិ ទិញទំនិញ ធ្វើការ មើលភាពយន្ត លេងកំសាន្ត ស្រាវជ្រាវ ។ល។ ដូច្នេះវាជារឿងសំខាន់ណាស់ដែលយើងត្រូវតែចេះធ្វើគេហទំព័រមួយឱ្យមានស្តង់ដារ។

យើងមកយល់ដឹងពីភាពខុសគ្នារវាងគេហទំព័រដែលមានស្តង់ដារ និងគ្មានស្តង់ដារថាតើវាមានផលប៉ះពាល់ដូចម្តេចចំពោះអ្នកប្រើប្រាស់។ ពួកយើងនឹងប្រៀបធៀបគេហទំព័រពីរផ្សេងគ្នា ដោយគេហទំព័រទាំងពីរមានគោលដៅ និងគោលបំណងដូចគ្នា ប្លែកត្រង់ថាគេហទំព័រមួយក្នុងចំណោមគេហទំព័រទាំងពីរគឺ មួយយកស្តង់ដារក្នុងការរចនាមកប្រើប្រាស់ ហើយមួយទៀតគឺអត់។ ដើម្បីស្វែងរកគេហទំព័រដែលយើងចង់ប្រៀបធៀប ដំបូងយើងត្រូវតែដឹងថាគេហទំព័រទាំងពីរមានគោលដៅនិងគោលបំណងដូចគ្នា កាលដែលយើងធ្វើដូច្នេះវាងាយស្រួលធ្វើឱ្យយើងសម្រេចចិត្ត និងងាយប្រៀបធៀប។ ទីពីរគឺពួកយើងចង់ស្វែងរកថាគេហទំព័រមួយណាមានស្តង់ដារក្នុងការរចនាហើយអត់នោះដើម្បីឱ្យយល់ដឹងកាន់តែច្បាស់ពួកយើងនឹងលើកឡើងគេហទំព័រពីរដែលលក់របងនៅលើប្រព័ន្ធអ៊ីនធឺណិតដូចគ្នាដោយពួកយើងក៏បានធ្វើការស្រាវជ្រាវផងដែរពីស្តង់ដារនៃការរចនា ហេតុនេះបានជាយើងអាចធ្វើការប្រៀបធៀបគេហទំព័រទាំងពីរនេះបាន។

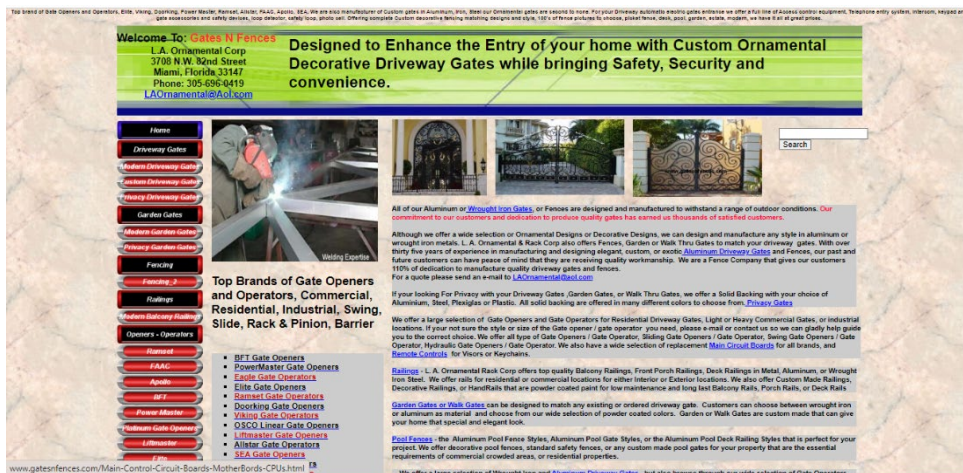
តោះយើងមកមើលគេហទំព័រទាំងពីរនេះជាមួយគ្នា៖

<https://www.mistergatesdirect.com/> MisterGatesDirect គឺជាគេហទំព័រលក់របងផ្ទះ។ គេហទំព័រនេះត្រូវបានធ្វើឡើងឆ្លងកាត់តាមស្តង់ដារនៃការចនាគេហទំព័រ។



(រូបភាពទី ០១ ៖ បង្ហាញពីគេហទំព័រ MisterGatesDirect)

ចំណែកឯ <http://www.gatesnfences.com/index.html> ដែលមានឈ្មោះគេហទំព័រ GateNFences ក៏ជាគេហទំព័រលក់របងដែរ តែគេហទំព័រនេះធ្វើឡើងមិនឆ្លងកាត់តាមស្តង់ដារនៃការចនាគេហទំព័រឡើយ។



(រូបភាពទី ០២ ៖ បង្ហាញពីគេហទំព័រ GateNFences)

២. បំណែងបញ្ជី

តើយើងត្រូវយល់ដឹងពីចំណុចអ្វីខ្លះទើបអាចធ្វើគេហទំព័រមួយប្រកបដោយស្តង់ដារ?

ពួកយើងសម្រេចជ្រើសយកសំណួរនេះមកស្រាវជ្រាវព្រោះគិតថាគេហទំព័រមួយលុះត្រាតែមានអ្នកចូលមកទស្សនាច្រើនក៏ដូចជាអាចទាក់ទាញអ្នកទស្សនាបានលុះត្រាតែធ្វើឡើងប្រកបតាមស្តង់ដារនៃការចនាគេហទំព័រ។

៣. គោលបំណងនៃការស្រាវជ្រាវ

ជាមួយនឹងការស្រាវជ្រាវនេះដែរពួកយើងមានបំណងចង់សម្រេចបាននូវចំណេះដឹង និងការយល់ដឹងជុំវិញការធ្វើអត្ថបទសារណា ហើយក៏ចង់ផ្តល់ព័ត៌មានដ៏មានតម្លៃទៅកាន់អ្នកអភិវឌ្ឍន៍គេហទំព័រផ្សេងទៀតដែលខ្វះខាតទៅលើចំណេះដឹងមួយនេះ។ បន្ទាប់ពីអានអត្ថបទសារណានេះចប់ ចំពោះនិស្សិត និងបងប្អូនដែលមានចំណង់ចំណូលចិត្តលើការសិក្សាមុខវិជ្ជាវចនាគេហទំព័រនេះនឹងទទួលបានចំណេះដឹងដ៏មានប្រយោជន៍ជាច្រើនដូចជា ផ្តល់ការគិតរកគំនិតក្នុងការវចនាគេហទំព័រមួយដែលងាយយល់ មិនស្មុគស្មាញថែមទាំងមានភាពទាក់ទាញនិងយល់កាន់តែច្បាស់ពីអ្វីដែលអ្នកប្រើប្រាស់ផ្តោតលើពេលដែលពួកគេកំពុងស្វែងរកគេហទំព័រ។ គោលបំណងនៃការលើកប្រធានបទនេះឡើងគឺដើម្បីចង់បញ្ជាក់ថាគេគោលការណ៍នៃការធ្វើគេហទំព័រឱ្យមានស្តង់ដារមួយមានសារៈសំខាន់យ៉ាងណាដល់អ្នកប្រើប្រាស់។

៤. ដេនកំណត់ និងវិសាលភាពនៃការស្រាវជ្រាវ

ចំពោះការសិក្សាស្រាវជ្រាវរបស់ពួកយើងទៅលើប្រធានបទខាងលើគឺបានផ្តោតទៅការសិក្សានៅក្នុងការបង្កើតឡើងគេហទំព័រមួយទៅតាមស្តង់ដារនៃការវចនា ហើយការសិក្សាស្រាវជ្រាវមួយនេះទៀតសោតគឺយើងខ្ញុំក៏បានបង្កើតជាគេហទំព័រភាពយន្តមួយដែលវាមានបង្កប់ជាការពន្យល់លម្អិតពីសារណាមួយនេះហើយគន្លឹះសំខាន់ៗដែលយើងខ្ញុំសរសេរក្នុងគេហទំព័រមានដូចជា៖

- អនុញ្ញាតឱ្យអ្នកប្រើប្រាស់ធ្វើការចុះឈ្មោះតាមរយៈអ៊ីមែល
- អនុញ្ញាតឱ្យអ្នកប្រើប្រាស់ស្វែងរកនូវចំណងជើងភាពយន្តដែលពួកគេចង់ទស្សនា
- អនុញ្ញាតឱ្យអ្នកប្រើប្រាស់ទស្សនាវីដេអូខ្លីៗរបស់ភាពយន្ត (Trailer)
- មានរៀបជាបញ្ជីនៃប្រភេទរបស់ភាពយន្តនៅខាងលើនៃគេហទំព័រ

៥. សារៈសំខាន់នៃការស្រាវជ្រាវ

តាមរយៈការប្រើប្រាស់គេហទំព័រមួយនេះ បានផ្តល់គុណសម្បត្តិដល់អ្នកប្រើប្រាស់ដូចជា៖

- ផ្តល់ការងាយស្រួលដល់អ្នកប្រើប្រាស់ មិនមានភាពស្មុគស្មាញ
- ផ្តល់ជាចំណេះដឹងដល់អ្នកអភិវឌ្ឍន៍គេហទំព័រផ្សេងទៀត ទាំងការប្រើប្រាស់ប្លង់ ពណ៌ ពុម្ពអក្សរ រូបភាព និងរូបតំណាង។

ជំពូកទី ១ ទ្រឹស្តី

១. អ្វីទៅជាគេហទំព័រ ?

គេហទំព័រ គឺជាបណ្តុំនៃបណ្តាញទំព័រ (web pages) និងមាតិកាផ្សេងៗដែលពាក់ព័ន្ធ (related contents) ហើយជាទូទៅគេហទំព័រត្រូវបានបង្កើតឡើងលើប្រធានបទជាក់លាក់ដូចជា ពាណិជ្ជកម្ម ព័ត៌មាន ជំនួញ ចំណេះដឹង ការអប់រំ ការកម្សាន្ត និងបណ្តាញសង្គម។ គេហទំព័រត្រូវបានកំណត់ដោយ Domain Name ហើយត្រូវបានដាក់ផ្សាយនៅលើ Web server យ៉ាងហោចណាស់មួយ មួយវិញទៀតអ្វីដែលខ្វះមិនបាននោះគឺអ៊ីនធឺណិត (Internet) ដែលជាចលករសំខាន់បំផុតនៃការដំណើរការរបស់គេហទំព័រ។

គេហទំព័របានផ្លាស់ប្តូរពិភពលោកទាំងស្រុងដោយបានផ្តល់នូវផលវិជ្ជមានជាច្រើនដល់ការរស់នៅប្រចាំថ្ងៃរបស់យើងដូចជា៖

- ការទំនាក់ទំនងគ្នាមានភាពងាយស្រួល និងលឿនជាងមុន
- ចំណេញពេលវេលា
- ងាយស្រួលដល់សហគ្រិន អាជីវករ និងអង្គភាពផ្សេងៗក្នុងការធ្វើទីផ្សារ
- ការជ្រាប ឬទទួលបានព័ត៌មានលឿនជាងមុន
- ការទិញដូរ និងលក់មានភាពងាយស្រួល
- ការរៀនសូត្រ និងសិក្សាដោយមិនចាំបាច់ទៅដល់សាលា ។ល។

១.១ អ្វីទៅជា www ?

WWW សំដៅទៅលើរាល់គេហទំព័រដែលអ្នកប្រើប្រាស់បានភ្ជាប់ ឬ access ទៅកាន់ Local Computer ឬឧបករណ៍ប្រើប្រាស់ផ្សេងៗតាមរយៈអ៊ីនធឺណិត។ ទំព័រ និងឯកសារនីមួយៗត្រូវបានភ្ជាប់គ្នាដោយ hyperlinked ដែលអ្នកប្រើប្រាស់ចុច ឬ click ដើម្បីស្វែងរកព័ត៌មានណាមួយ ដែលព័ត៌មានទាំងនោះ វាអាចជាសំឡេង រូបភាព អត្ថបទ និងវីដេអូ។

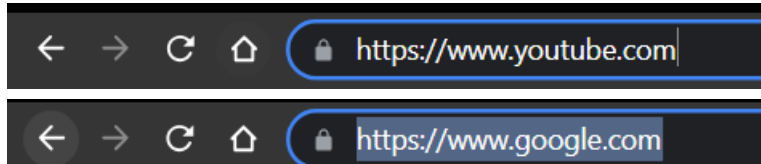
WWW ត្រូវបានបង្កើតឡើងដោយអ្នកវិទ្យាសាស្ត្រកុំព្យូទ័រជនជាតិអង់គ្លេសម្នាក់គឺ លោក Tim Berners-Lee។ លោកបង្កើតវាឡើងដោយសារតែធុញទ្រាន់ជាមួយនឹងការផ្ទុកឯកសារ (storing) ការធ្វើបច្ចុប្បន្នភាព (updating) និងការស្វែងរកឯកសារនានានៅក្នុងក្រុមហ៊ុនដែលតែងតែមានការផ្លាស់ប្តូរឥតឈប់ឈរ។

១.២ អ្វីទៅជា Domain Name ?

Domain name គឺជាឈ្មោះមួយដែលត្រូវបានស្គាល់ថាជាអាសយដ្ឋានអ៊ីនធឺណិត (internet address) របស់គេហទំព័រ។ គេប្រើប្រាស់វាដើម្បីបញ្ជាក់នូវធនធានរបស់អ៊ីនធឺណិត (internet resources) ដូច

ជា កុំព្យូទ័រ បណ្តាញ network និងសេវាកម្ម។ មួយវិញទៀតវាក៏ត្រូវបានគេប្រើប្រាស់ដើម្បីបញ្ជាក់អត្តសញ្ញាណឱ្យ IP Address ផងដែរ។ អ្នកប្រើប្រាស់នឹងមានភាពងាយស្រួលក្នុងការស្វែងរកគេហទំព័ររបស់យើងនៅពេលដែលគេហទំព័រនោះមាន Domain name។

ឧទាហរណ៍៖ <https://www.youtube.com/> , <https://www.google.com/>



១.៣ អ្វីទៅជា Internet ?

អ៊ីនធឺណិតគឺជាប្រព័ន្ធសកលនៃការភ្ជាប់គ្នារបស់បណ្តាញកុំព្យូទ័រដែលប្រើប្រាស់ protocol (TCP/IP) នៃអ៊ីនធឺណិតដើម្បីបង្កើតទំនាក់ទំនងរវាងបណ្តាញ Networks និងឧបករណ៍ប្រើប្រាស់នានា។ វាបានធ្វើបដិវត្តទៅលើការទំនាក់ទំនង និងវិធីសាស្ត្រនៃពាណិជ្ជកម្មដោយអនុញ្ញាតឱ្យកុំព្យូទ័រទាំងអស់នៅជុំវិញពិភពលោកអាចទាក់ទងគ្នាទៅវិញទៅមក។ ជាមួយគ្នានេះដែរ អ៊ីនធឺណិតក៏អាចភ្ជាប់ស្ទើរតែគ្រប់ព័ត៌មាន គ្រប់ទិន្នន័យ ហើយក៏អាចឱ្យយើងទាក់ទងជាមួយអ្នកផ្សេងទៀតក្នុងពិភពលោកផងដែរ។

ចាប់តាំងពីមានវត្តមានអ៊ីនធឺណិត ពិភពលោកត្រូវបានផ្លាស់ប្តូរទាំងស្រុង មិនថាជាជនជាតិបាល វិស័យជំនួញ វិស័យអប់រំ និងវិស័យសុខាភិបាល។ វាបានក្លាយទៅជាកូនសោរដ៏សំខាន់មួយនៃការវិវត្តន៍របស់សង្គម។ មិនតែប៉ុណ្ណោះការធ្វើប្រាស្រ័យទាក់ទងគ្នាគឺកាន់តែងាយស្រួល មិនថាអ្នកនៅទីណា កន្លែងណា ឬពេលណា ឱ្យតែអ្នកមានអ៊ីនធឺណិតអ្នកនឹងអាចធ្វើការទាក់ទងគ្នាទៅវិញទៅមកបានយ៉ាងងាយ។

អ៊ីនធឺណិតត្រូវបានបង្កើតសម្រាប់ធ្វើជាអាវុធរបស់រដ្ឋាភិបាលនៅក្នុងសង្គ្រាមត្រជាក់ដើម្បីឱ្យអ្នកវិទ្យាសាស្ត្រ និងអ្នកស្រាវជ្រាវអាចទាក់ទង និងចែករំលែកទិន្នន័យជាមួយភាគីម្ខាងទៀត។ ទម្រង់ដើមនៃអ៊ីនធឺណិតគឺ ARPANET ដែលត្រូវបានបង្កើតឡើងដើម្បីរចែករំលែកធនធាននានា។ ការបង្កើតឡើងនេះក៏មានការពាក់ព័ន្ធ និងចូលរួមពីសំណាក់អ្នកវិទ្យាសាស្ត្រ និងវិស្វកម្មជាច្រើន ដោយពួកគេមានចក្ខុវិស័យលើអ៊ីនធឺណិតជាយូរណាស់មកហើយ មុនពេលដែលបច្ចេកវិទ្យាក្នុងការបង្កើតវាត្រូវបានរកឃើញទៅទៀត។ គ្រោងការណ៍ដំបូងរបស់អ៊ីនធឺណិតបានមកដល់នៅដើមទសវត្សរ៍ឆ្នាំ ១៩៦០ ហើយក្រោយមកក្រុមអ្នកវិទ្យាសាស្ត្រកុំព្យូទ័របានបង្កើតវិធីសាស្ត្រមួយក្នុងការបញ្ជូន ឬផ្ទេរទិន្នន័យអេឡិចត្រូនិចដែលវិធីសាស្ត្រនោះត្រូវបានហៅថា packet switching ។

២. ការរចនាគេហទំព័រ (Web Design)

ការរចនាគេហទំព័រ គឺសំដៅទៅលើការរចនាទំព័រដែលបង្ហាញនៅលើអ៊ីនធឺណិត។ វាជាទូទៅសំដៅទៅលើបទពិសោធន៍ដែលអ្នកប្រើប្រាស់ទទួលបានជាងការអភិវឌ្ឍន៍ប្រព័ន្ធ Software។ កាលដើមឡើយការរចនាគេហទំព័រនេះ គឺធ្វើតម្រូវតែអេក្រង់កុំព្យូទ័រ (Desktop browsers) តែប៉ុណ្ណោះ ក្រោយមកនៅពាក់កណ្តាលឆ្នាំ 2010 ការរចនាសម្រាប់ឧបករណ៍ផ្សេងៗទៀតដូចជា Tablet និងទូរស័ព្ទក៏ត្រូវបានចាត់ទុកថាជាការសំខាន់ដែរ។

ក្នុងដំណើរការនៃការរចនាគេហទំព័រ អ្នករចនាមានរឿងច្រើនដែលត្រូវខ្វល់ដូចជា អ្វីដែលអ្នកប្រើប្រាស់មើលឃើញ (appearance) ការតម្រៀប ឬរៀបចំប្លង់ (layout) ពុម្ពអក្សរ រូបភាព និងពណ៌ជាដើម។ មួយវិញទៀតអ្វីដែលសំខាន់បំផុតនោះគឺ ការរចនាដែលមាន responsive ព្រោះកាលណាគេហទំព័រមួយដែលមាន responsive នោះគេហទំព័រនឹងអាចបត់បែនបានទៅតាមទំហំអេក្រង់នីមួយៗនៃឧបករណ៍ប្រើប្រាស់មិនថាកុំព្យូទ័រ tablet និងទូរស័ព្ទឡើយ។ ដែលទាំងអស់នេះស្ថិតនៅក្នុងគោលបំណងតែមួយគឺ អារម្មណ៍ និងបទពិសោធន៍របស់អ្នកប្រើប្រាស់ ជាហេតុដែលអ្នករចនាដ៏ល្អម្នាក់តែងតែប្រើប្រាស់ចិត្តកាយ ដាក់អារម្មណ៍ និងដើរតួជាអ្នកប្រើប្រាស់ផ្ទាល់ដើម្បីសម្រេចបាននូវគេហទំព័រដ៏ល្អមួយ។ គេហទំព័រដ៏ល្អមួយជាគេហទំព័រ ដែលធ្វើឱ្យអ្នកប្រើប្រាស់មានអារម្មណ៍ថា មិនស្មុគស្មាញ ងាយស្រួលអាន និងងាយយល់។

២.១ ចំណុចប្រទាក់អ្នកប្រើប្រាស់ (UI = User Interface)

ចំណុចប្រទាក់អ្នកប្រើប្រាស់ ឬ User interface ដែលទូទៅគេនិយមហៅថា UI ជាចំណុចដែលផ្ដោតទៅលើការរចនាភាពស្រស់ស្អាតរបស់គេហទំព័រ ឬ application មួយ។ បើនិយាយទាក់ទងទៅនឹងការរចនាគេហទំព័រ UI មានតួនាទីយ៉ាងសំខាន់ទៅលើ Appearance, Interaction, Usability, User Behavior និង Feel។

- Appearance សំដៅទៅលើភាពស្រស់ស្អាតរបស់គេហទំព័រមួយ វាជាចំណុចដែលអ្នករចនាគេហទំព័រត្រូវតែខ្វាយខ្វល់ ព្រោះចំណុចនេះក៏អាចជាឆ្នាក់ទាក់ទាញឱ្យគេចង់ប្រើប្រាស់គេហទំព័ររបស់យើងដែរ។
 - Interactivity សំដៅទៅលើចលនាដែលកើតមានបន្ទាប់ពីអ្នកប្រើប្រាស់បាន click ដាក់ mouse ពីលើ ឬធ្វើការផ្លាស់ប្តូរផ្សេងៗ។ វាធ្វើចលនាដោយផ្លាស់ប្តូរពីទម្រង់មួយទៅទម្រង់មួយផ្សេងទៀត អាចជាការប្តូរពណ៌អក្សរ ប្តូរពណ៌ប៊ូតុង ។ល។
- ឧទាហរណ៍៖



- Usability សំដៅទៅលើភាពងាយស្រួលប្រើប្រាស់នៃ features នីមួយៗ។ ជាធម្មតានៅពេលដែលគេហទំព័រមួយមានភាពងាយអាន ងាយយល់ មិនស្មុគស្មាញ នឹងធ្វើឱ្យគេនិយមប្រើប្រាស់ ឬ visit វា។
- User Behavior សំដៅទៅលើថាតើអ្នកប្រើប្រាស់ចំណាយពេលប៉ុន្មានទៅលើផលិតផលរបស់យើង ហើយផលិតផលមួយណាដែលពួកគេចុច ឬ Click ញឹកជាងគេ។ល។ វាផ្ដោតទៅលើទំនាក់ទំនងរវាងផលិតផលរបស់យើង និងអ្នកប្រើប្រាស់។
- Feel សំដៅទៅលើអារម្មណ៍ដែលអ្នកប្រើប្រាស់ទទួលបានកំឡុងពេល និងបន្ទាប់ពីចូលទៅគេហទំព័ររបស់យើង។

២.២ បទពិសោធន៍អ្នកប្រើប្រាស់ (UX = User Experience)

បទពិសោធន៍អ្នកប្រើប្រាស់ ឬ User Experience ដែលទូទៅគេនិយមហៅថា UX។ បើនិយាយពីការរចនាគេហទំព័រ វាផ្ដោតពិសេសទៅលើការផ្តល់បទពិសោធន៍ទៅដល់អ្នកប្រើប្រាស់ ដោយសំដៅលើភាពងាយស្រួលប្រើប្រាស់នៃគេហទំព័រនោះ។

បើនិយាយឱ្យស៊ីជម្រៅរវាង UX និង UI ដែលយើងបានបកស្រាយពីខាងលើ គឺទាំងពីរនេះពួកវាធ្វើការបំពេញគ្នាទៅវិញទៅមកដោយមិនអាចខ្វះមួយណាបានឡើយ។ ចង្វាក់ការងារនៅក្នុងដំណើរការរចនាគេហទំព័រមួយ UX និង UI មានរៀងៗខ្លួន។ UX មានតួនាទីកំណត់អត្តសញ្ញាណនៃតម្រូវការរបស់អ្នកប្រើប្រាស់ និងអភិវឌ្ឍផែនការមួយដើម្បីបំពេញតម្រូវការទាំងនោះ។ ឯ UI មានតួនាទីក្នុងការបង្កើតនូវសក់ណ្ឌភាព និងអារម្មណ៍នៃគេហទំព័រ ដែលរួមបញ្ចូលទាំង Layout និងក្រាហ្វិក។ UX/UI បានបង្កើតនូវបទពិសោធន៍ដល់អ្នកប្រើប្រាស់ដ៏ជិតស្និទ្ធ និងមានប្រសិទ្ធភាពមួយ។

៣ ការរចនាក្រាហ្វិក (Graphic Design)

Graphic Design ឬការរចនាក្រាហ្វិកគឺជាទម្រង់នៃទំនាក់ទំនងមួយដែលសំដៅទៅលើពណ៌ រាង រូបភាព និងពាក្យពេជន៍ដែលគេប្រើដើម្បីបង្កើតនូវមាតិកាសម្រាប់ print ឬដាក់ផ្សព្វផ្សាយតាមប្រព័ន្ធឌីជីថលផ្សេងៗ។ ចំពោះការរចនាគេហទំព័រវិញ ការរចនាក្រាហ្វិកមានតួនាទីក្នុងការបង្កើតគំនិត រូបភាព និងធាតុផ្សេងៗនៃគេហទំព័រ។ វាគឺជាឧបករណ៍ ឬ tool មួយសម្រាប់បង្កើននូវប្រសិទ្ធភាពនៃគេហទំព័រ ក៏ដូចជាសារទៅកាន់អ្នកប្រើប្រាស់។

នេះគឺជាអ្វីដែលការរចនាក្រាហ្វិកបានជួយបំពេញក្នុងចង្វាក់នៃការរចនាគេហទំព័រមួយ៖

- រចនា logo, icon, banner, ប៊ូតុង និងក្រាហ្វិកផ្សេងៗទៀតសម្រាប់កំណត់កំណត់អត្តសញ្ញាណនៃគេហទំព័រ។
- ជ្រើសរើសព័ត៌មានអក្សរ ពណ៌ ប្លង់ និងរូបភាពដែលសាកសមនឹងរចនាបទ និងគោលបំណងរបស់គេហទំព័រ។
- បង្កើត mockups, wireframes, prototypes និង user flow ដើម្បីបង្ហាញនូវមុខងាររបស់គេហទំព័រ និងបទពិសោធន៍អ្នកប្រើប្រាស់។
- ការងារខ្លះការប្រើប្រាស់ជាមួយកម្មវិធីរចនាដូចជា Adobe Photoshop, Illustrator ឬ InDesign ដើម្បីបង្កើតធាតុផ្សេងៗ។

៣.១ ប្លង់ (Layout)

នៅក្នុងការរចនាក្រាហ្វិក ប្លង់ ឬ Layout មានតួនាទីក្នុងការរៀបចំប្រព័ន្ធនៃធាតុដែលនៅលើទំព័រ។ វាទាក់ទងទៅនឹងការរៀបចំអក្សរ រូបភាព និងធាតុក្រាហ្វិកផ្សេងៗទៀត ដើម្បីឱ្យអ្នកប្រើងាយស្រួលក្នុងការមើល និងអានជាដើម។ ការរៀបចំប្លង់ដែលល្អមួយ នឹងអាចជួយឱ្យទំនាក់ទំនងមានប្រសិទ្ធភាព និងអាចបង្ហាញអ្វីដែលអ្នកចង់បង្ហាញទៅអ្នកប្រើប្រាស់បាន។

ទាំងនេះជាគោលការណ៍មូលដ្ឋានដែលជួយសម្លៀករូបអ្នកក្នុងការរៀបចំប្លង់ ឬ Layout ៖

- ភាពនៅជិតគ្នា (Proximity): ការដាក់ធាតុដែលពាក់ព័ន្ធជាក្រុម (Group related items) ដើម្បីបង្កើតនូវការភ្ជាប់រវាងពួកវា។
- ការតម្រឹម (Alignment): តម្រឹមធាតុឱ្យស្មើគ្នាដើម្បីឱ្យមើលឃើញស្រឡះភ្នែក និងរៀបរយ។
- ការប្រើប្រាស់ធាតុដដែលៗ (Repetition): ការប្រើប្រាស់ធាតុមួយម្តងហើយម្តងទៀត ដើម្បីបង្កើតបានជា texture ឬ វាយនភាព។
- កម្រិតពណ៌ (Contrast): បង្កើនភាពលេចធ្លោនៃធាតុ ដែលងាយនឹងចាប់អារម្មណ៍។
- គម្លាតពណ៌ស (White space): ប្រើប្រាស់គម្លាតពណ៌ស ដើម្បីឱ្យការរចនារបស់យើងមើលទៅមានតុល្យភាព។

៣.២ អក្សរ (Typography)

នៅក្នុងការរចនាក្រាហ្វិក ការវាយអក្សរ ឬ Typography គឺជាសិល្បៈនៃការរៀបចំសរសេរឃ្លា ឬប្រយោគឱ្យមើលទៅងាយស្រួលអាន ទាក់ទាញ និងមានអត្ថន័យ។ Typography ពាក់ព័ន្ធនឹងការជ្រើសរើស និងរួមបញ្ចូលគ្នានៃព័ត៌មានអក្សរ ទំហំអក្សរ ពណ៌ និងការដកឃ្លា ដើម្បីឱ្យមើលទៅជាប្រភពមួយដែលមានភ្ជាប់

ជាមួយសារទៅកាន់អ្នកប្រើប្រាស់។ លើរាល់ការងារដូចជា សៀវភៅ poster គេហទំព័រ និង logo សុទ្ធតែមានការដាក់បញ្ចូលនៃសិល្បៈនៃការរៀបចំអក្សរ។

ធាតុផ្សេងៗដែលអ្នកត្រូវដឹងនៅក្នុងសិល្បៈនៃការប្រើប្រាស់អក្សរ៖

- Style: រាល់អក្សរទាំងអស់មានច្រើនសណ្ឋានទាំងរូបរាង និងរចនាបទ ដែលអក្សរទាំងនោះមានលក្ខណៈពិសេសរៀងៗខ្លួន។
- Font: ពុម្ពអក្សរគឺជាក្រុមនៃតួអក្សរដែលបិទនៅក្រោមរចនាបទដូចគ្នា ទំហំដូចគ្នា និងកម្រាស់ដូចគ្នា។
- Kerning: ជាការកែសម្រួលចន្លោះរវាងតួអក្សរនីមួយៗដែលវាជួយឱ្យមើលទៅងាយស្រួលអាន។
- Leading: ទាក់ទងនឹងគម្លាតនៃបន្ទាត់ក្នុងអត្ថបទដែលអាចឱ្យបន្ទាត់ប្រយោគមានលក្ខណៈរង្វើល ឬប្រកៀកគ្នា។
- Tracking: ជាការកែសម្រួលចន្លោះរវាងអក្សរទាំងអស់នៅក្នុងអត្ថបទ ដែលអាចឱ្យតួអក្សរនីមួយៗមានលក្ខណៈរង្វើល ឬប្រកៀកគ្នា។
- Hierarchy: វាផ្ដោតទៅលើលំដាប់នៃភាពសំខាន់របស់អក្សរដូចជា ចំណងជើង ចំណងជើងរង តួអត្ថបទ ។ល។ វាជួយអ្នកអានងាយយល់ និងងាយអាន។

៣.៣ ពណ៌ (Color)

ពណ៌ គឺជាធាតុដ៏សំខាន់ក្នុងការរចនាក្រាហ្វិក ពីព្រោះវាអាចផ្តល់ជាសារ បង្កើតអារម្មណ៍ និងបញ្ចេញអារម្មណ៍។ ទ្រឹស្តីនៃពណ៌ គឺជាក្របខណ្ឌ ឬ framework ដែលពន្យល់ពីរបៀបដែលមនុស្សយល់ឃើញពីពណ៌ របៀបលាយ ផ្គូផ្គង និងដាក់មិត្តនៃពណ៌ទៅវិញទៅមក។ វាណែនាំពីការប្រើប្រាស់ពណ៌នៅក្នុងទម្រង់នៃសិល្បៈ និងការរចនាដើម្បីផ្តល់ជាសារទាំងស្រោកណាភាព និងទំនោរអារម្មណ៍។

អ្នករចនាបានប្រើប្រាស់ទ្រឹស្តីនៃពណ៌ដើម្បីជ្រើសរើសពណ៌ចម្រុះសម្រាប់ចំណុចប្រទាក់ដែលជាសមត្ថភាពអុបទិករបស់មនុស្ស ចិត្តវិទ្យា រូបវន្ត និងច្រើនទៀត។ ពណ៌មានច្រើនប្រភេទផ្សេងៗគ្នាដូចជា monochromatic, analogous, complementary, split-complementary, tradic និង tetradic។ ពណ៌នីមួយៗមានគុណសម្បត្តិ និងគុណវិបត្តិផ្ទាល់ខ្លួន អាស្រ័យលើបរិបទដែលទាមទារ។

៤ ការអភិវឌ្ឍន៍គេហទំព័រ (Web Development)

ចាប់តាំងពីចុងទសវត្សរ៍ឆ្នាំ ៩០មកប្រព័ន្ធអ៊ីនធឺណិតបានផ្លាស់ប្តូររបៀបដែលមនុស្សទទួលបានព័ត៌មានហើយការអភិវឌ្ឍន៍គេហទំព័រក៏បន្តវិវត្តន៍ទៅមុខជានិច្ច។ សារៈសំខាន់ចម្បងរបស់វាចំពោះប្រភព Digital

របស់យើងនាំឱ្យតម្រូវការចំពោះអ្នកអភិវឌ្ឍន៍គេហទំព័រមានការទាមទារខ្ពស់។ Web Development ដែលយើង អាចហៅថាភាសាខ្មែរបានថាការអភិវឌ្ឍន៍គេហទំព័រ វាសំដៅទៅលើការងារដែលជាប់ទាក់ទងជាមួយនឹង ការបង្កើត អភិវឌ្ឍ និងថែរក្សាគេហទំព័រដែលដំណើរការនៅលើប្រព័ន្ធអ៊ីនធឺណិតក៏ដូចជា browser ជាដើម។ នៅខាងក្នុងការអភិវឌ្ឍន៍គេហទំព័រមានរួមបញ្ចូលទាំង ការចនាគេហទំព័រ (Web Design) កម្មវិធីក្នុងការបង្កើត គេហទំព័រ (Web Programming) និងការគ្រប់គ្រងមូលដ្ឋានទិន្នន័យ (Database Management)។ ការ អភិវឌ្ឍន៍គេហទំព័រគឺទាក់ទងយ៉ាងជិតស្និទ្ធទៅនឹងការងារនៃការបង្កើតលក្ខណៈពិសេស និងមុខងាររបស់កម្មវិធី (ការចនាគេហទំព័រ)។ ពាក្យថាអភិវឌ្ឍន៍ជាធម្មតាត្រូវបានប្រើក្នុងការសាងសង់ក៏ដូចជាកែតម្រូវរបស់អ្វីមួយឱ្យមាន ភាពជឿនលឿន និងឆ្លោះទៅមុខជានិរន្តរ៍។ ជាក់ស្តែងការសរសេរកម្មវិធីនៃគេហទំព័រ គឺជាវិធីសាស្ត្រដែលគេយក មកប្រើប្រាស់ដើម្បីអភិវឌ្ឍន៍គេហទំព័រឱ្យមានភាពជឿនលឿនតាមសម័យកាលឬនិន្នាការនៃប្រព័ន្ធអ៊ីនធឺណិត។

ការអភិវឌ្ឍន៍គេហទំព័រត្រូវបានចែកចេញជាពីរប្រភេទសំខាន់ៗគឺ៖

- ការអភិវឌ្ឍន៍ផ្នែកខាងមុខនៃគេហទំព័រ (Front-End Development)
- ការអភិវឌ្ឍន៍ផ្នែកខាងក្រោយនៃគេហទំព័រ (Back-End Development)

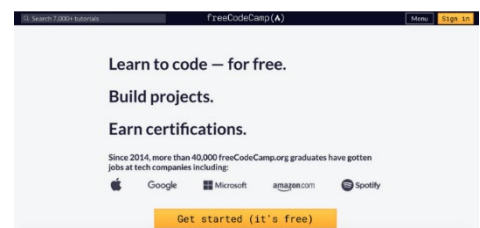
ដោយការសរសេរសារណារបស់ពួកយើងគឺលើកឡើងពីការសរសេរគេហទំព័រមួយឱ្យមានស្តង់ដារ ដូច្នេះប្រធានបទមួយនេះវាជាប់ទាក់ទងទាំងការអភិវឌ្ឍន៍ផ្នែកខាងមុខនៃគេហទំព័រ (Front-End Development) និងការអភិវឌ្ឍន៍ផ្នែកខាងក្រោយនៃគេហទំព័រ (Back-End Development) តិចតួចហេតុនេះពួកយើងនឹងលើក យកការអភិវឌ្ឍន៍ទាំងពីរនេះមកបកស្រាយ។

មូលដ្ឋានគ្រឹះដែលជាប់ទាក់ទងនឹងការអភិវឌ្ឍន៍គេហទំព័រគឺភាសាសរសេរកម្មវិធីដែលមានដូចជា HTML , CSS , JavaScript , Bootstrap , PHP , React JS និង jQuery & Ajax ។

៤.១ ការអភិវឌ្ឍន៍គេហទំព័រផ្នែកខាងមុខ (Front-end Development)

អ្វីៗទាំងអស់ដែលអ្នកឃើញនៅលើគេហទំព័រ ដូចជា ប៊ូតុង (Button) តំណភ្ជាប់ (Link) រូបភាពដែល មានចលនា (Animation) ហើយនិងរឿងច្រើនទៀតគឺសុទ្ធតែត្រូវបានបង្កើតឡើងមកដោយអ្នកអភិវឌ្ឍន៍ គេហទំព័រផ្នែកខាងមុខ។ ការងារនេះជាការងាររបស់ពួកគេដោយទទួលយកចក្ខុវិស័យក៏ដូចជាគំនិតក្នុងការចនា ពីអតិថិជន ហើយអនុវត្តតាមរយៈការសរសេរកូដ។ ឥឡូវយើងមកមើលទំព័រខាងមុខនៃគេហទំព័រ freeCo- deCamp ៖

(រូបភាពទី ០៤៖ គេហទំព័រ freeCodeCamp)



គ្រប់យ៉ាងដែលនៅលើទំព័រនេះ ចាប់តាំងពី logo ទៅដល់រូបភាពស្វែងរក ប៊ូតុង ឬរូបរាងទាំងមូល ហើយនិងសកម្មភាពផ្សេងៗទៀតដែលអ្នកប្រើប្រាស់អាចភ្ជាប់ជាមួយបានគឺសុទ្ធតែត្រូវបានបង្កើតឡើងដោយអ្នកអភិវឌ្ឍន៍គេហទំព័រផ្នែកខាងមុខមិនត្រឹមតែប៉ុណ្ណឹងទេអ្នកអភិវឌ្ឍន៍គេហទំព័រផ្នែកខាងមុខគឺត្រូវតែច្បាស់ថាគេហទំព័រគឺសុទ្ធតែអាចប្រើប្រាស់ និងមើលទៅត្រូវស្តង់ដារអេក្រង់នៃឧបករណ៍ប្រើប្រាស់ទាំងអស់ (Phones, Tablets, Computer Screens)។

ភាសាសំខាន់ៗចំនួនបីដែលអ្នកត្រូវតែយល់ដឹងឱ្យបានច្បាស់ដើម្បីអភិវឌ្ឍន៍គេហទំព័រផ្នែកខាងមុខឱ្យបានស្អាត និងត្រូវតាមស្តង់ដារមានដូចជា៖

- HTML
- CSS
- JavaScript ។



(រូបភាពទី ០៤ ៖ Logo នៃភាសា JavaScript HTML និង CSS)

៤.១.១ អ្វីទៅជា HTML ?

HTML តំណាងឱ្យពាក្យ Hyper Text Markup Language គឺជាភាសាសម្គាល់ស្តង់ដារដែលកម្មវិធីរុករកតាមអ៊ីនធឺណិតប្រើដើម្បីបកស្រាយ សរសេរអត្ថបទ បង្ហាញរូបភាពក៏ដូចជាសម្ភារៈផ្សេងទៀតទៅក្នុងគេហទំព័រដែលអាចមើលឃើញ ឬអាចស្តាប់បាន។ HTML ត្រូវបានបង្កើតឡើងដំបូងនៅក្នុងឆ្នាំ ១៩៨០ ដោយលោក Tim Berners-Lee ហើយរហូតមកដល់បច្ចុប្បន្ន HTML គឺមាន ៤ ជំនាន់ទៅហើយដោយមាន HTML 2, HTML 3, HTML 4 និង HTML 5។

ជាធម្មតាភាសាមួយនេះត្រូវបានគេសរសេរដោយភ្ជាប់ជាមួយនឹង CSS និងភាសាដូចជា JavaScript ជាដើមដើម្បីធ្វើឱ្យគេហទំព័រមានភាពទាក់ទាញ ក៏ដូចជាផ្តល់លក្ខណៈពិសេសផ្សេងៗដល់អ្នកប្រើប្រាស់។ បច្ចុប្បន្ន HTML អាចវិវត្តន៍ជាមួយនឹងកំណែទម្រង់ជំនាន់ថ្មីដែលជាជំនាន់ទី 5 (HTML 5 ដែលវាមានសមត្ថភាពគាំទ្រវីដេអូ និងសម្លេងដែលនេះជាលក្ខណៈពិសេសមួយរបស់វា។

៤.១.២ អ្វីទៅជា CSS ?

CSS គឺជាពាក្យកាត់របស់ពាក្យ Cascading Style Sheets ដែលជាភាសាកម្មវិធីដូចគ្នាទៅនឹង HTML ផងដែរ ដែលគ្រាន់តែជាប្រភេទឯកសារមួយបែបដែលគេប្រើដើម្បីកែច្នៃរូបរាង (ពុម្ពអក្សរ, ពណ៌, ទំហំ...) នៃមាតិកានៅលើគេហទំព័រ បើនិយាយមួយបែបទៀត CSS គឺជាសម្លៀកបំពាក់ដែលគេស្រោបពីលើ HTML ដើម្បីឱ្យគេហទំព័រមានភាពស្រស់ស្អាត ក៏ដូចជាទាក់ទាញ។ CSS នេះទៀតសោត គឺត្រូវបានដាក់ណែនាំឱ្យប្រើ ដំបូងដោយលោក Hakon Wium Lie នៅក្នុងឆ្នាំ ១៩៩៤ ដែលត្រូវជាមិត្តរួមការងាររបស់លោក Tim Berners-Lee នៅក្នុងក្រុមហ៊ុន CERN ដែលជាអង្គការអឺរ៉ុបដែលស្រាវជ្រាវគ្រាប់បែកនុយក្លេអ៊ែរ។ មិនខុសពី HTML នោះ ទេ CSS ក៏ត្រូវបានវិវត្តន៍មកជាច្រើនជំនាន់ផងដែរ ដែលមាន CSS 1, CSS 2, CSS 2.1, និង CSS 3 ហើយ ក៏ត្រៀមនឹងបញ្ចេញជំនាន់ទី 4 (CSS 4) នៅពេលអនាគតផងដែរ។

លក្ខណៈពិសេសនៃជំនាន់ចុងក្រោយរបស់ CSS (CSS3) គឺមាននូវ Syntax ក៏ដូចជា Feature ថ្មីៗ ជាច្រើនដែលនឹងអាចជួយឱ្យគេហទំព័រមានភាពស្រស់ស្អាត និងងាយស្រួលសម្រាប់អ្នកប្រើប្រាស់។

៤.១.៣ អ្វីទៅជា Bootstrap ?

Bootstrap ដែលមានឈ្មោះដើមថា Twitter Blueprint គឺត្រូវបានរកឃើញ និងបង្កើតឡើងនៅក្នុងឆ្នាំ ២០១១ ដោយលោក Mark Otto និង លោក Jacob Thornton ដែលពួកគាត់ទាំងពីរគឺជាបុគ្គលិករបស់ក្រុមហ៊ុន Twitter។ Bootstrap គឺជាប្រភេទ framework ដែលល្បីល្បាញបំផុតដោយវាអនុញ្ញាតឱ្យអ្នកអភិវឌ្ឍន៍គេហទំព័រ ក៏ដូចជាមនុស្សទូទៅទាញយកទៅប្រើប្រាស់ដោយមិនគិតថ្លៃសម្រាប់រចនាគេហទំព័រ។ វាគឺត្រូវបាន គេចាត់ទុកថាជាបណ្ណាល័យមួយរបស់ HTML, CSS និង JavaScript ដែលសម្រួលដល់ការអភិវឌ្ឍន៍គេហទំព័រ។ មកដល់បច្ចុប្បន្ន Bootstrap គឺបានឈានមកដល់ជំនាន់ទីប្រាំហើយ ដែលត្រូវបានគេស្គាល់ជា Bootstrap 5 ដោយជំនាន់មុនៗគឺចាប់ផ្តើមឡើងដោយជំនាន់ទី ២, ៣, ៤ និងចុងក្រោយបំផុតគឺទី៥។

Bootstrap v5.2 ត្រូវបានដាក់ប្រកាសឱ្យប្រើប្រាស់ក្នុងខែមេសា ឆ្នាំ២០២២ រួមជាមួយនឹងការបន្ថែម មុខងារមួយចំនួនបន្ទាប់ពី Bootstrap v5.0។ រហូតមកដល់បច្ចុប្បន្ននេះ Bootstrap ជំនាន់ចុងក្រោយគឺ Bootstrap v5.3 ដែលបានប្រកាសដាក់ឱ្យប្រើប្រាស់នៅក្នុងខែឧសភា ឆ្នាំ២០២៣។



(រូបភាពទី ០៥ ៖ Logo របស់ Bootstrap)

៤.១.៣ អ្វីទៅជា Font-Awesome ?

Font-Awesome គឺជាកញ្ចប់ពុម្ពអក្សរ និង icon មួយប្រភេទដែលគេប្រើដើម្បីជាជំនួយបន្ថែមទៅដល់ CSS និង Less។ ដោយវាត្រូវបានបង្កើតឡើងដំបូងនៅក្នុងឆ្នាំ ២០១២ ដោយលោក Dave Gandy ដើម្បីប្រើជាមួយនឹង Bootstrap។ តាមការស្រង់ស្ថិតិក្នុងឆ្នាំ ២០២០ ដោយគេហទំព័រ Wikipedia កម្មវិធី Font-Awesome គឺត្រូវបានយកមកប្រើប្រាស់ដើម្បីបង្កើតគេហទំព័រចំនួន ៣៨% នៃកម្មវិធីរុករកទាំងអស់នៅលើអ៊ីនធឺណិតដែលធ្វើឱ្យវាស្ថិតនៅលេខពីរតាមពីក្រោយ Google Fonts ដែលត្រូវបានគេប្រើប្រាស់ច្រើនបំផុត។

លក្ខណៈពិសេសរបស់ Font-Awesome គឺវាអនុញ្ញាតឱ្យអ្នកប្រើប្រាស់យកវាទៅប្រើដើម្បីដាក់រូបតំណាង (icon ដែលមានមាត្រដ្ឋានអាចកែសម្រួលបាន ក៏ដូចជាអាចប្តូរទំហំ ពណ៌ ស្រមោល និងអ្វីទាំងអស់នៃពុម្ពអក្សរដោយផ្អែកលើ CSS។ សព្វថ្ងៃ Font-Awesome គឺមានដល់ជំនាន់ទីប្រាំមួយហើយដែលជាជំនាន់ចុងក្រោយបំផុតត្រូវបានដាក់ឱ្យប្រើប្រាស់នៅខែកុម្ភៈ ឆ្នាំ ២០២២។



(រូបភាពទី ០៦ ៖ Logo របស់ Font-Awesome)

៤.២ ការអភិវឌ្ឍន៍គេហទំព័រផ្នែកខាងក្រោយ (Back-end Development)

ការអភិវឌ្ឍន៍គេហទំព័រផ្នែកខាងក្រោយមានន័យថាជាការធ្វើការនៅលើកម្មវិធីផ្នែកខាងម៉ាស៊ីនមេដែលវាផ្តោតទៅលើអ្វីទាំងអស់ដែលអ្នកមិនអាចមើលឃើញនៅលើគេហទំព័រ។ អ្នកអភិវឌ្ឍន៍ផ្នែកខាងក្រោយមានភារកិច្ចត្រូវធានាថាគេហទំព័រដំណើរការបានត្រឹមត្រូវ ដោយផ្តោតទៅលើមូលដ្ឋានទិន្នន័យ APIs ស្ថាបត្យកម្មនៃគេហទំព័រ និងផ្សេងៗទៀតដែលពាក់ព័ន្ធនឹងម៉ាស៊ីនមេ។ ពួកគេប្រើកូដដែលជួយឱ្យកម្មវិធីរុករកទាក់ទងជាមួយមូលដ្ឋានទិន្នន័យ រក្សាទុក យល់ និងលុបចោលទិន្នន័យ។

ទាំងនេះគឺជាមូលដ្ឋានទូទៅដែលអ្នកអភិវឌ្ឍន៍គេហទំព័រផ្នែកខាងក្រោយត្រូវធ្វើប្រចាំថ្ងៃ៖

- បង្កើត និងថែទាំគេហទំព័រ
- សរសេរកូដដែលមានគុណភាពខ្ពស់
- អនុវត្តការធ្វើតេស្តធានាគុណភាព (QA)
- វាយតម្លៃប្រសិទ្ធភាព និងល្បឿន
- ដោះស្រាយបញ្ហា និងបំបាត់កំហុស

- ហ្វីកហាត់ និងគាំទ្រ។

ភាសាកម្មវិធីសំខាន់ៗមួយចំនួនដែលចាំបាច់ចំពោះការអភិវឌ្ឍន៍គេហទំព័រផ្នែកខាងក្រោយ៖

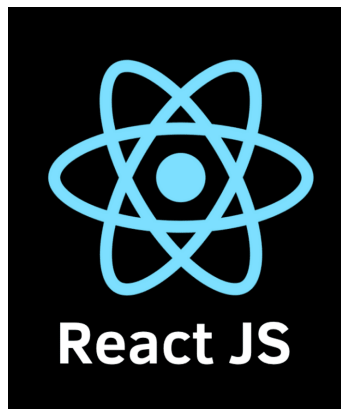
- Python
- PHP
- JavaScript
- Ruby
- Java
- C# ។

៤.២.១ អ្វីទៅជា JavaScript ?

JavaScript ដែលមានពាក្យកាត់ថា JS គឺជាភាសាកម្មវិធីស្នូលមួយដែលគេយកមកប្រើជាមួយ HTML និង CSS។ វាត្រូវបានបញ្ចេញឱ្យប្រើប្រាស់ដំបូងបំផុតនៅឆ្នាំ ១៩៩៥ ដោយលោក Brendan Eich ហើយត្រូវបានអភិវឌ្ឍប្រើនៅក្នុងកម្មវិធីរុករក Netscape 2។ ពីរឆ្នាំក្រោយមក វាត្រូវបានគេទទួលស្គាល់ជាភាសាអន្តរជាតិបែបស្តង់ដារ ECMAScript ចាប់តាំងពីពេលនោះវាត្រូវបានយកមកប្រើប្រាស់ដើម្បីអភិវឌ្ឍន៍កម្មវិធីរុករកតាមអ៊ីនធឺណិតជាច្រើនដូចជា (Firefox, Chrome, Internet Explorer... ។ គិតត្រឹមឆ្នាំ ២០២២ JavaScript គឺត្រូវបានគេយកមកសរសេរដើម្បីបង្កើតគេហទំព័រច្រើនជាងគេបំផុតដែលមានរហូតទៅដល់ ៩៨%។ រាល់កម្មវិធីរុករកតាមអ៊ីនធឺណិតធំៗទាំងអស់គឺសុទ្ធតែមានម៉ាស៊ីន JavaScript ពិសេសដើម្បីប្រតិបត្តិកូដនៅលើឧបករណ៍អ្នកប្រើប្រាស់។ JavaScript ក៏អាចចាត់ទុកថាជាភាសាដ៏ស្មុគស្មាញមួយផងដែរ ដែលវាមិនគាំទ្រជាមួយភាសាផ្សេងទៀតក្រៅពីខ្លួនវាឡើយ នោះមានន័យថាបើសិនជាអ្នកកំពុងសរសេរកម្មវិធីមួយដោយប្រើ JavaScript នោះតម្រូវឱ្យអ្នកប្រើប្រាស់ភាសាមួយនេះតែមួយគត់។ ការពិតទៅ JavaScript មានរចនាសម្ព័ន្ធដូចគ្នាទៅនឹងភាសា C អញ្ចឹង គ្រាន់តែវាមានភាសាខុសប្លែកពីគេ ប៉ុន្តែដូចគេត្រង់ការដំណើរការ ប្រតិបត្តិការកូដ។ បើយើងមកពិចារណាពីភាសាកម្មវិធីដ៏ដូចជាការសាងសង់ផ្ទះអញ្ចឹងដែរ ដោយ HTML ជាអ្នកធ្វើទ្រង់ទ្រាយស្ថាបត្យកម្មនៃផ្ទះ ចំណែក CSS ជាអ្នកក្រាលកម្រាលព្រំ និងលាបពណ៌ជញ្ជាំង ដើម្បី តុបតែងផ្ទះ ចុងក្រោយ JavaScript ជាអ្នកបន្ថែមនូវធាតុបន្ទាប់បន្សំនៅក្នុងផ្ទះដូចជាការបើកទ្វារ និងដាក់ ភ្លើង បំភ្លឺជាដើម។ ពិតណាស់អ្នកក៏អាចបង្កើតគេហទំព័រមួយដោយគ្មាន JavaScript បានផងដែរ ប៉ុន្តែគុណវិបត្តិគឺថាគេហទំព័ររបស់អ្នកនឹងមើលទៅដូចជាគេហទំព័រស្ថិតក្នុងឆ្នាំ ១៩៩៥ អញ្ចឹង។

៤.២.២ អ្វីទៅជា React JS ?

React ដែលគេស្គាល់ថាជា React JS ឬ React.js គឺជាបណ្ណាល័យ JavaScript front-end ឥតគិតថ្លៃនិងបើកចំហរឱ្យយកទៅប្រើប្រាស់សម្រាប់បង្កើតចំណុចប្រទាក់អ្នកប្រើប្រាស់ដោយផ្អែកលើសមាសធាតុ (Component ហើយវាអាចដំណើរការទៅបានដោយសរសេរជាជំនួយនៅក្នុង JavaScript ដោយមានន័យថាវាជា Framework មួយដែលជួយបន្ថែមភាពរស់រវើកដល់ JavaScript។ React គឺត្រូវបានដាក់ឱ្យប្រើប្រាស់ដោយបើកចំហរនៅក្នុងឆ្នាំ ២០១៣ ឡើងដោយលោក Jordan Walke ដែលជាបុគ្គលិកនៅក្នុងក្រុមហ៊ុន Meta ហើយវាត្រូវបានបញ្ចេញសាកល្បងក្នុងការបង្កើត News Feed នៅលើ Facebook កាលពី ឆ្នាំ ២០១១ ដោយសារតែការសាកល្បងទទួលបានការជោគជ័យគេក៏ប្រើវាបង្កើត News Feed នៅលើ Instagram បន្ថែមទៀតនៅឆ្នាំ ២០១២។ សព្វថ្ងៃ React គឺត្រូវបានគ្រប់គ្រងដោយក្រុមហ៊ុន Meta។ អ្នកអភិវឌ្ឍន៍ជាច្រើនប្រើ React ដើម្បីបង្កើតកម្មវិធីបណ្តាញផ្សេងៗដោយសារតែវាជួយឱ្យទិន្នន័យ ផ្លាស់ប្តូរដោយមិនចាំបាច់ដំណើរការគេហទំព័រម្តងទៀតឡើយ។ គោលដៅស្នូលនៃ React គឺវាបំបាត់ និង សាមញ្ញ។ វាត្រូវបានគេយកប្រើជាមួយនឹងបណ្ណាល័យ JavaScript ដទៃទៀតដូចជា frameworks, Angular នៅក្នុង MVC ជាដើម។ React គឺត្រូវបានវិវត្តន៍មកចំនួន ៤៧ជំនាន់មកហើយ ដោយជំនាន់ចុងក្រោយបំផុត React v18.2 គឺត្រូវបានបញ្ចេញនៅក្នុងថ្ងៃទី ១៤ ខែ មិថុនា ឆ្នាំ ២០២២។



(រូបភាពទី ០៧ ៖ Logo របស់ React JS)

៤.២.៣ អ្វីទៅជា JQuery និង Ajax ?

jQuery គឺជាបណ្ណាល័យរបស់ JavaScript ដែលអនុញ្ញាតឱ្យអ្នកប្រើប្រាស់អាចសរសេរកម្មវិធីបានដោយមិនត្រូវបានរំខានដោយព័ត៌មានស្មុគស្មាញជាច្រើននៃកម្មវិធី JavaScript ឡើយ។ jQuery ត្រូវបានបង្កើតឡើងដំបូងនៅក្នុងខែមករា ឆ្នាំ២០០៦ ដោយលោក John Resig ដែលលោកបានយកលំនាំពីបណ្ណាល័យ cssQuery របស់លោក Dean Edward។ គិតត្រឹមខែសីហា ឆ្នាំ២០២២ jQuery ត្រូវបានប្រើប្រាស់ ៧៧% ដោយគេហទំព័រល្បីល្បាញចំនួន ១០លានគេហទំព័រ។ កិច្ចការមួយចំនួនធំដែលត្រូវសរសេរនៅក្នុង JavaScript

មានកូដជាច្រើនជូន តែត្រូវបានហៅដោយកូដមួយបន្ទាត់ប៉ុណ្ណោះចំពោះ jQuery។ មិនតែប៉ុណ្ណោះភាសាកូដមួយនេះមិនមែនជាភាសាដែលដាច់ដោយឡែកពីគេទេដោយវាអាចភ្ជាប់ជាមួយភាសា JavaScript បានថែមទាំងអាចជួយសម្រួលភាពស្មុគស្មាញពី JavaScript ដូចជាការរៀបចំ DOM និងការហៅ AJAX ជាដើម។

Ajax មានពាក្យពេញថា Asynchronous JavaScript XML គឺជាការផ្ទុកទិន្នន័យនៅក្នុង Background ហើយបង្ហាញវិញនៅលើគេហទំព័រ Web Page ដោយមិនបាច់ដំណើរការគេហទំព័រទាំងមូលឡើងវិញឡើយ (Reloading ។ ពាក្យមួយនេះត្រូវបានដាក់ឱ្យប្រើប្រាស់ជាសាធារណៈនៅថ្ងៃទី ១៨ ខែកុម្ភៈ ឆ្នាំ២០០៥ ដោយលោក Jesse James Garret។ កម្មវិធីដែលប្រើ AJAX មានដូចជា Gmail, Google Maps, YouTube និង Facebook ជាដើម។ វាក៏ជាបច្ចេកទេសមួយដែលគេប្រើដើម្បីជួយឱ្យ Dynamic Web Page ដំណើរការបានលឿន។ Ajax រួមបញ្ចូលនូវឧបករណ៍កម្មវិធីជាច្រើនដូចជា DHTML, CSS, XML, Server Side Scripting PHP, DOM និង Microsoft object។



(រូបភាពទី ០៨ ៖ Logo របស់ jQuery និង Ajax)

៤.២.៤ អ្វីទៅជា PHP ?

តើអ្នកដឹងទេ PHP មានពាក្យពេញថា Hypertext Preprocessor គឺជា ភាសាកម្មវិធីមួយបែប ដែលប្រើសម្រាប់ប្រតិបត្តិការគេហទំព័រ។ វាគឺជាភាសាប្រភេទបើកចំហមានន័យថាដាក់ឱ្យប្រើដោយសេរី ហើយនេះក៏ជាហេតុផលដែលជួយឱ្យ PHP គឺជាភាសាដ៏ល្បីល្បាញមួយ សម្រាប់ការអភិវឌ្ឍន៍ គេហទំព័រ និងរួមចំណែកក្នុងការអភិវឌ្ឍន៍ Platform ផ្សេងៗដូចជា WordPress, Facebook, Google, Yahoo! ជាដើម។ PHP ត្រូវបានបង្កើតឡើងដំបូងដោយអ្នកសរសេរកម្មវិធីម្នាក់ឈ្មោះថា Rasmus Lerdorf នៅក្នុងឆ្នាំ ១៩៩៣ ហើយបញ្ចេញឱ្យប្រើប្រាស់ជាសកលនៅក្នុងឆ្នាំ ១៩៩៥។ PHP ជាប្រភេទភាសាដែល ប្រើទៅ លើម៉ាស៊ីនមេ (Server-Side) ដែលនេះមានន័យថាវាប្រតិបត្តិការតែនៅលើម៉ាស៊ីនណាដែលបានតម្លើង Hard Drive មករួចរាល់ហើយតែប៉ុណ្ណោះ។ ចំណែកភាសាដែលប្រើតែទៅលើខាងអ្នកប្រើប្រាស់ (Client-Side) ដូចជា JavaScript ដើរតួនាទី

ជាមុខងារនៅក្នុងកម្មវិធីរុករករបស់អ្នកប្រើប្រាស់តែប៉ុណ្ណោះ។ ហេតុផលនេះហើយបានជាការដោះស្រាយបញ្ហាដំណើរការនៃគេហទំព័រពាក់ព័ន្ធនឹងការធ្វើការជាមួយ PHP។

PHP ត្រូវបានវិវត្តន៍មកជាច្រើនជំនាន់ហើយដោយចាប់តាំងពីជំនាន់ទី 3 4 5 6 7 និង PHP 8 គឺត្រូវបានបញ្ចេញជាផ្លូវការនៅថ្ងៃទី ២៦ ខែវិច្ឆិកា ឆ្នាំ ២០២០ ដែលវាបានក្លាយទៅជាជំនាន់ចុងក្រោយបំផុតនៃ PHP។



(រូបភាពទី ០៩ ៖ Logo របស់ jQuery និង Ajax)

ជំពូកទី ២ ស្ថានភាពទូទៅ

១. វិធីសាស្ត្រស្រាវជ្រាវ

១.១. ការជ្រើសរើសប្រធានបទស្រាវជ្រាវ

ការជ្រើសរើសប្រធានបទពឹងផ្អែកទៅលើកត្តាផ្សេងៗជាច្រើន ដែលក្នុងនោះរួមមានកម្រិត លក្ខណៈ និងទំហំជាដើម។

ប្រធានបទត្រូវបានបែងចែកជា ៤ កម្រិតគឺ៖

- កម្រិតមូលដ្ឋានសំដៅទៅលើកម្រិតភូមិ ឃុំ សង្កាត់ ដែលមានទំហំតូច និងងាយស្រួលក្នុងការស្រាវជ្រាវ
- កម្រិតខេត្តសំដៅទៅលើកម្រិតប្រជុំនៃស្រុកជាច្រើន ដែលមានទំហំមធ្យមក្នុងការស្រាវជ្រាវ
- កម្រិតថ្នាក់ជាតិសំដៅទៅលើកម្រិតទូទាំងប្រទេស
- កម្រិតអន្តរជាតិសំដៅទៅលើថ្នាក់តំបន់ដូចជា អាស៊ាន អង្គការសហប្រជាជាតិ កិច្ច សហប្រតិបត្តិការអន្តរជាតិ។ល។

របៀបក្នុងការស្វែងរកប្រធានបទមានច្រើនវិធីសាស្ត្រដូចជា៖

- ឯកសារតាមសៀវភៅ និងតាមបណ្តាញអ៊ីនធើណែត
- ចំណង់ចំណូលចិត្ត
- ប្រធានបទនៃសារណាចាស់ៗ
- សារព័ត៌មាន កាសែត ឬទស្សនាវដ្តី
- គ្រូជីកនាំ
- បញ្ហាចោទក្នុងសង្គម ឬព្រឹត្តិការណ៍ថ្មីៗ។ល។

ដោយសារតែការសិក្សាស្រាវជ្រាវនេះជាកម្រិតថ្នាក់បរិញ្ញាបត្រដូចនេះប្រធានបទដែលបានជ្រើសរើសឡើងស្ថិតនៅក្នុងកម្រិតមូលដ្ឋាន។ អាស្រ័យទៅតាមចំណង់ចំណូលរបស់ក្រុមយើងខ្ញុំ និងបានមើលឃើញពីបញ្ហាចោទដែលកើតមានឡើងក្នុងសង្គម ទាក់ទងនឹងកង្វះចំណេះដឹងក្នុងការចនាគេហទំព័រឱ្យមានស្តង់ដារដែលអាចផ្តល់ភាពងាយស្រួលដល់អ្នកប្រើប្រាស់ ទើបបណ្តាលឱ្យក្រុមយើងខ្ញុំសម្រេចជ្រើសរើសប្រធានបទ ការចនាគេហទំព័រ នេះឡើង។ ទន្ទឹមនឹងនេះ បើគ្មានការផ្តល់ជាយោបល់ និងណែនាំពីលោកគ្រូ ក្រុមរបស់យើងខ្ញុំក៏មិនច្បាស់ថា ប្រធានបទដែលជ្រើសរើសឡើងមកនេះត្រឹមត្រូវតាមការកំណត់បានដែរ។

១.២. ការប្រមូលទិន្នន័យ

សម្រាប់ការសិក្សាស្រាវជ្រាវ ការប្រមូលទិន្នន័យពិតជាមានសារៈសំខាន់ខ្លាំងណាស់ព្រោះនេះបញ្ជាក់ពី ភាពច្បាស់លាស់ និងមិនល្មើចបន្លំនៃការស្រាវជ្រាវដែលបានធ្វើឡើង។ ហេតុដូច្នេះ ទិន្នន័យដែលប្រមូលបាន ត្រូវច្បាស់ថាមានប្រភពច្បាស់លាស់ដូចជា ឈ្មោះអ្នកនិពន្ធ គ្រឹះស្ថានស្រាវជ្រាវ ឬបោះពុម្ព ឆ្នាំសរសេរ និងកម្ម សិទ្ធិបញ្ញាជាដើម។ វិធីសាស្ត្រដែលការស្រាវជ្រាវរបស់យើងបានប្រើប្រាស់ក្នុងការប្រមូលទិន្នន័យគឺ៖

- ទិន្នន័យដែលប្រើប្រាស់ភាគច្រើនចេញពីសៀវភៅដែលមានឈ្មោះអ្នកនិពន្ធ គ្រឹះស្ថាន ស្រាវជ្រាវ ឬបោះពុម្ព ដែលមានឆ្នាំសរសេរច្បាស់លាស់
- ទិន្នន័យដែលប្រើប្រាស់បន្ទាប់បន្សំគឺចេញពីគេហទំព័រដែលមានឈ្មោះអ្នកនិពន្ធ និង កាលបរិច្ឆេទសរសេរច្បាស់លាស់ដូចគ្នា
- ជាចុងក្រោយគឺជាការប្រមូលទិន្នន័យពីអ្នកដែលប្រើប្រាស់គេហទំព័រផ្ទាល់។

១.៣. ការកំណត់តម្រូវការព័ត៌មាន

ចំពោះទិន្នន័យដែលប្រមូលបានក៏មិនមែនប្រាកដថាសុទ្ធតែអាចប្រើប្រាស់បានទាំងអស់ដែរ។ ដូចនេះ យើងត្រូវសិក្សាមើលថាទិន្នន័យណាដែលទាក់ទង និងមានសារៈសំខាន់សម្រាប់ប្រធានបទតែប៉ុណ្ណោះ ទើបអាច លើកយកមកប្រើប្រាស់ និងដាក់បញ្ចូលបាន។

១.៤. ការរៀបចំរចនាសម្ព័ន្ធនៃប្រព័ន្ធគេហទំព័រ

ដោយសារចង់ឱ្យអ្នកដែលអានសៀវភៅក៏ដូចជា មើលសារណាមួយនេះមានការយល់ដឹងឱ្យកាន់តែ ច្បាស់ពីប្រធានបទដែលបានសរសេរឡើងមក ហេតុនេះទើបយើងខ្ញុំចង់ណែនាំពីការរៀបចំរចនាសម្ព័ន្ធនៃប្រព័ន្ធ គេហទំព័រ គឺធ្វើឡើងប្រកបដោយភាពទាក់ទាញ រស់រវើក សាមញ្ញ និងងាយស្រួលក្នុងការប្រើប្រាស់។ ខាងក្រោម នេះជាការបកស្រាយពីផ្នែកនីមួយៗ៖

- ផ្នែក Code Editor ត្រូវបានសរសេរឡើងដោយប្រើប្រាស់នូវកម្មវិធី Visual Studio Code ដោយសារតែវាមានភាពងាយស្រួលក្នុងការប្រើប្រាស់មានមុខងារច្រើន និងអាច បង្កើតជា User Interface ដ៏ទាក់ទាញ។
- ផ្នែក User Interface ត្រូវបានសរសេរឡើងដោយប្រើប្រាស់ភាសាដូចជា៖ HTML, CSS, Bootstrap, JavaScript។
- ផ្នែក API ត្រូវបានទាញចេញយកមកប្រើប្រាស់ពីគេហទំព័រភាពយន្ត TMDB ។
- ផ្នែក Web Browser សំដៅដល់កន្លែងប្រើប្រាស់គេហទំព័រ គឺអាចប្រើប្រាស់ដូចជា Google Chrome, Opera និង Firefox។

១.៥. ការសរសេរកូដ

ការសរសេរកូដ ជាដំណាក់កាលមួយដែលប្រើប្រាស់ពេលវេលាជាងគេ និងមានភាពស្មុគស្មាញជាងគេ។ ចំពោះភាសាដែលត្រូវប្រើប្រាស់ក្នុងការសរសេរកូដ គឺត្រូវបានរៀបរាប់នៅក្នុងជំពូកទី១។ ឯការសរសេរកូដនេះ ទៀតសោតមិនមែនសម្រាប់តែក្នុងគោលបំណងឱ្យចេញជាលទ្ធផលតែប៉ុណ្ណោះទេ ប៉ុន្តែរាល់តម្រូវការ និងភាពខ្វះចន្លោះនានាដែលអ្នកប្រើប្រាស់គេហទំព័រតែងលើកឡើង យើងក៏បានខិតខំសរសេរកូដដើម្បីទៅបំពេញបន្ថែមចន្លោះប្រហោងទាំងនោះផងដែរ។

១.៦. ការធ្វើតេស្តសាកល្បង និងការថែទាំរក្សាគេហទំព័រ

ការធ្វើតេស្តសាកល្បងសំដៅលើការស្វែងរកនូវកំហុស ភាពខ្វះចន្លោះ និងបញ្ហានានាដែលកើតមានក្នុងការសរសេរកូដនេះឡើង ដើម្បីយើងអាចធ្វើការកែសម្រួល និងបំពេញបន្ថែមមកវិញ។ ការធ្វើតេស្តសាកល្បងនេះចែកចេញជា ២គឺ៖

- ការតេស្តដោយផ្ទាល់
- ការតេស្តដោយអ្នកប្រើប្រាស់ដទៃ។

២. គម្រោងពេលវេលានៃការស្រាវជ្រាវ

គម្រោងពេលវេលានៃការស្រាវជ្រាវនេះសំដៅដល់ការបែងចែកពេលវេលាដើម្បីបង្កើតជាកិច្ចការស្រាវជ្រាវនេះឡើង។ ដោយក្នុងតារាងមួយនេះគឺក្រុមយើងនឹងបង្ហាញពីថេរវេលាក៏ដូចជាកាលបរិច្ឆេទនៃខណៈពេលក្នុងការធ្វើសកម្មភាពសារណាចាប់តាំងពីការប្រមូលទិន្នន័យ ឡើងគម្រោង ដំណើរការសរសេរកូដ សរសេរសារណា តេស្តសាកល្បង ប្រគល់សារណា និងរហូតដល់ការការពារសារណា។

ខាងក្រោមនេះជាតារាងបង្ហាញពីពេលវេលាក្នុងការធ្វើសារណានេះឡើង៖

ល.រ	តារាងបែងចែកពេលវេលា	សកម្មភាពដែលសម្រេចបាន
១	១០/០៤/២០២៣ - ២៥/០៤/២០២៣	ប្រមូលទិន្នន័យ
២	២៦/០៤/២០២៣ - ៥/០៥/២០២៣	ឡើងគម្រោង
៣	០៦/០៥/២០២៣ - ១០/០៦/២០២៣	ដំណើរការសរសេរកូដ
៤	០៧/០៦/២០២៣ - ៣០/០៦/២០២៣	សរសេរសារណា
៥	២០/០៧/២០២៣ - ២៧/០៧/២០២៣	តេស្តសាកល្បង
៦	២៨/០៧/២០២៣ - ២១/០៨/២០២៣	ប្រគល់សារណា
៧	១៣/០៩/២០២៣	ការការពារសារណា

ជំពូកទី ៣ លទ្ធផលនៃការស្រាវជ្រាវ

១. ដំណើរការនៃការរចនាគេហទំព័រ

ចំពោះគេហទំព័រដែលពួកយើងបានរចនាយើងឡើងដើម្បីចាត់ទុកជាគំរូ ឬឧទាហរណ៍ដើម្បីឱ្យប្រិយមិត្តអ្នកអានកាន់តែយល់ពីប្រធានបទរបស់ពួកយើងដែលនិយាយអំពី **រចនាគេហទំព័រ** នោះត្រូវបានបង្កើតឡើងដោយឆ្លងកាត់នូវទ្រឹស្តីដូចជា បទពិសោធន៍អ្នកប្រើប្រាស់ ចំណុចប្រទាក់អ្នកប្រើប្រាស់ ការប្រើប្រាស់ប្លង់ ពណ៌ និងអក្សរបានយ៉ាងត្រឹមត្រូវ។ ជាមួយគ្នានេះដែរគេហទំព័រទាំងមូលក៏ត្រូវបានអភិវឌ្ឍន៍ដោយប្រើប្រាស់ភាសាកូដជាច្រើនដូចជា HTML CSS Bootstrap JavaScript JavaScript XML និង ReactJS។ ពួកយើងក៏បានប្រើ software មួយចំនួនដូចជា Visual Studio Adobe Illustrator និង Adobe Photoshop ដើម្បី សរសេរ និងរចនាធាតុផ្សេងៗនៅលើគេហទំព័រ។

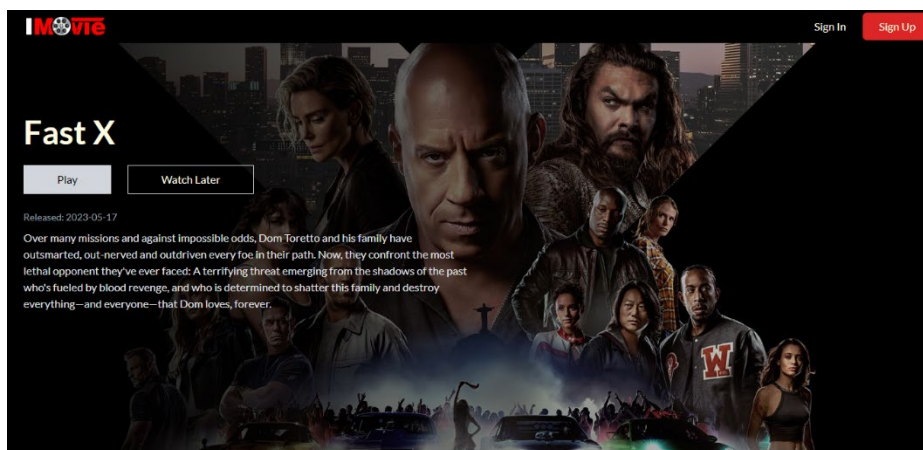
ដូចដែលអ្វីដែលយើងបញ្ជាក់ខាងលើថា គេហទំព័រទស្សនាភាពយន្តដែលយើងរចនាឡើងនេះគ្រាន់តែជាគេហទំព័រគំរូ ឬឧទាហរណ៍ ដូច្នេះពួកយើងមិនបានបន្ថែមមុខងារដែលអាចឱ្យអ្នកប្រើប្រាស់ទស្សនាខ្សែភាពយន្តពេញបាននោះទេ ដែលអាចត្រឹមតែទស្សនាឈុតខ្លីៗ ឬ Trailer នៃខ្សែភាពយន្តតែប៉ុណ្ណោះ។

២. ការរចនាលើផ្ទាំងរចបសាយ

២.១. របៀប Login និង Register នៅកាន់រចបសាយ

ដើម្បីអាចចូលទៅកាន់វេបសាយបានដំបូងយើងត្រូវ ៖

- បើក Browser
- វាយអាសយដ្ឋានរបស់គេហទំព័រ



(រូបភាពទី ១០ ៖ បង្ហាញពីផ្ទាំងដំបូងគេហទំព័រ)

- ធ្វើការ Sign up និង Sign in ដោយប្រើប្រាស់ អ៊ីមែលនិងលេខសម្ងាត់



Sign InSign Up

Sign Up

Email

Password

Sign Up

☐ Remember me

Need Help?

Already subscribed to IMovie? [Sign In](#)

(រូបភាពទី ១១ ៖ បង្ហាញពី Sign Up Form)

IMOVIE

Sign InSign Up

Sign In

Email

Password

Sign In

☐ Remember me

Need Help?

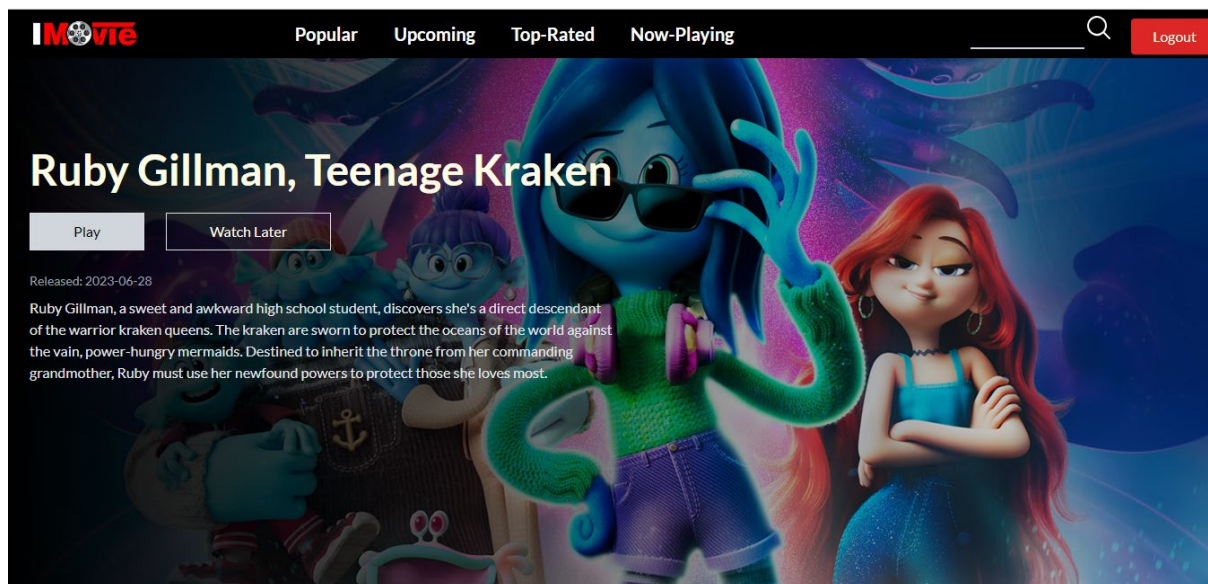
New to IMovie? [Sign Up](#)

(រូបភាពទី ១២ ៖ បង្ហាញពី Sign In Form)

២.២. ផ្ទៃខាងក្នុងនៃគេហទំព័រ

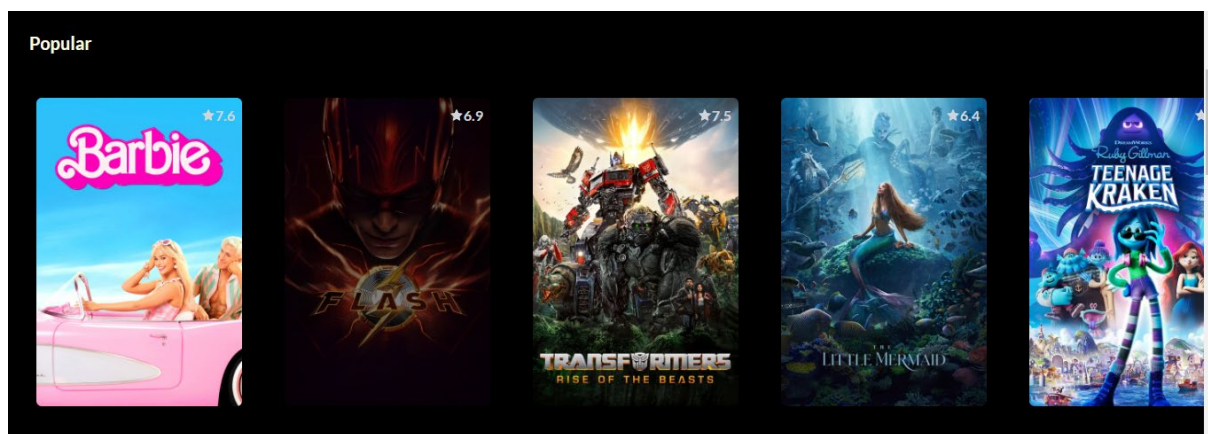
បន្ទាប់ពីធ្វើការ Sign in ឬក៏ Sign up រួចរាល់ហើយនិងចូលមកដល់ផ្ទៃខាងក្នុងនៃគេហទំព័រដែល មានលក្ខណៈពិសេសដូចជា៖

- មានរៀបជាប្រភេទនៃភាពយន្តនៅលើផ្ទាំងខាងលើរបស់គេហទំព័រ និង ប៊ូតុងសម្រាប់ស្វែងរកភាពយន្ត



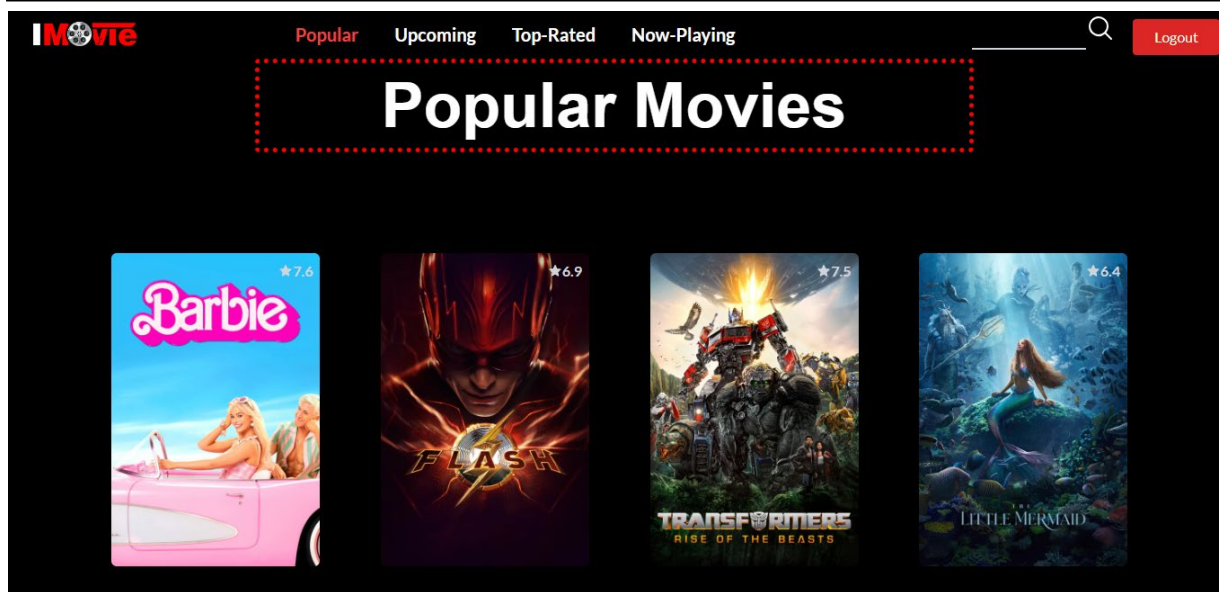
(រូបភាពទី ១៣ ៖ បង្ហាញពីផ្ទៃខាងក្នុងនៃគេហទំព័រ)

- មានបង្ហាញជាភាពយន្តមួយចំនួនទៅតាមប្រភេទរបស់សាច់រឿង

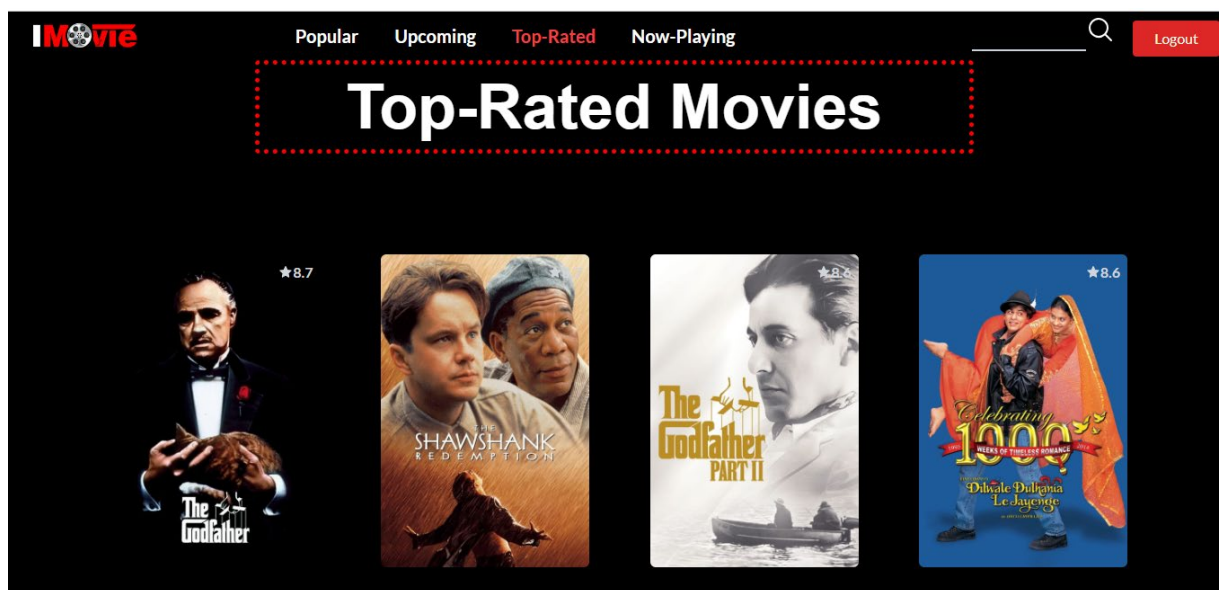


(រូបភាពទី ១៤ ៖ បង្ហាញពី Slide នៃខ្សែភាពយន្តនៅលើ Home page)

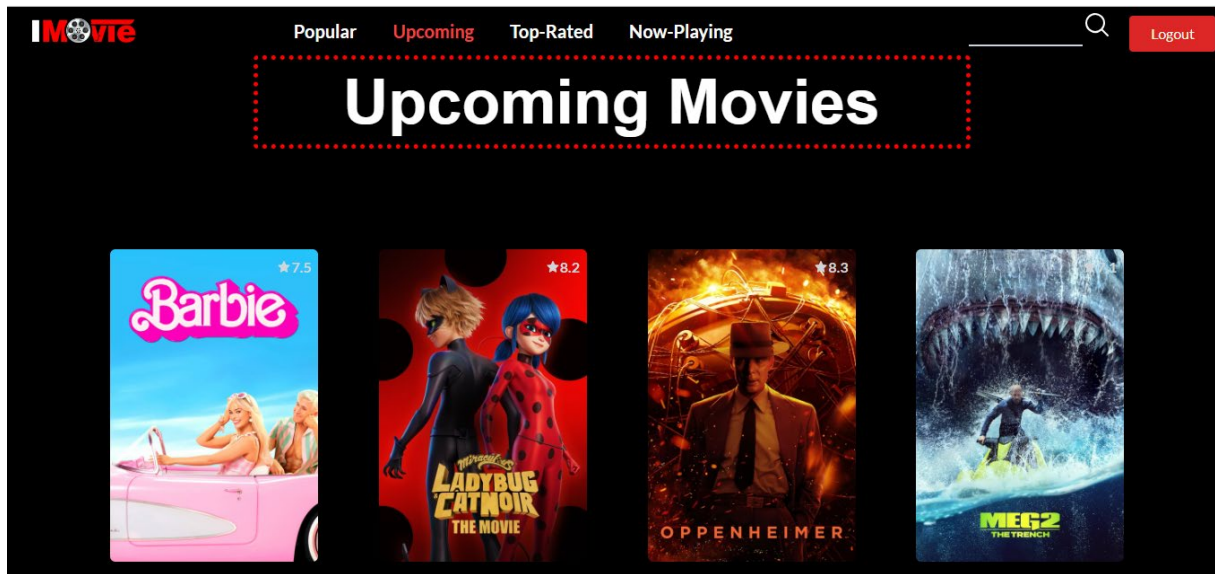
- ប្រភេទនៃភាពយន្តត្រូវបានរៀបចំតាមមាតិការបស់សាច់រឿង



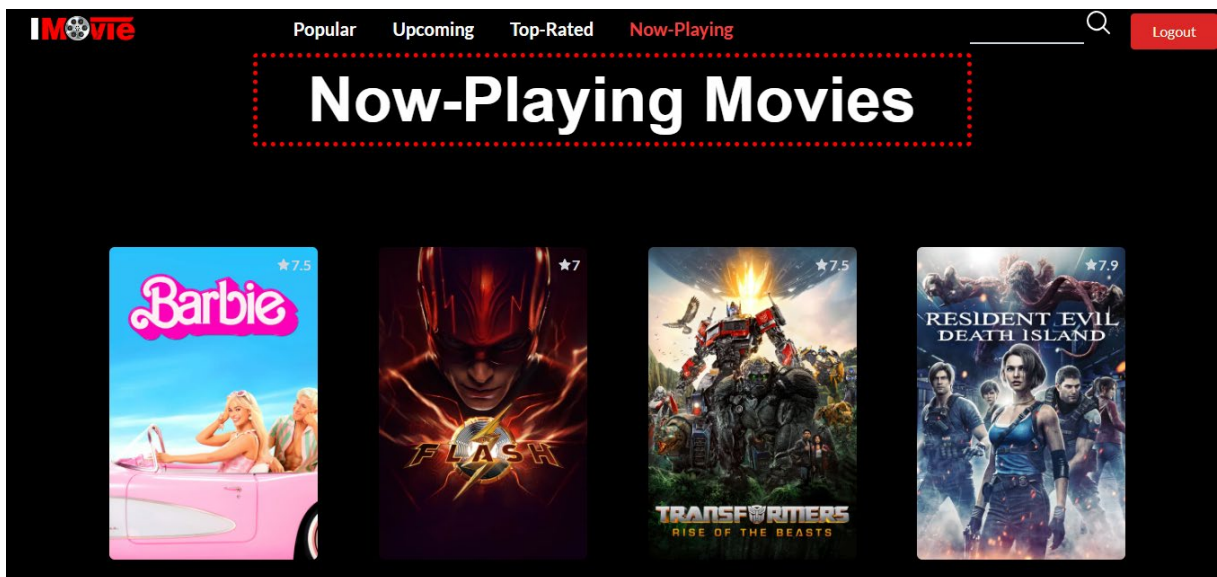
(រូបភាពទី ១៥ ៖ បង្ហាញពីខ្សែភាពយន្តដែលកំពុងល្បី)



(រូបភាពទី ១៦ ៖ បង្ហាញពីខ្សែភាពយន្តដែលមានការ RATE ខ្ពស់)



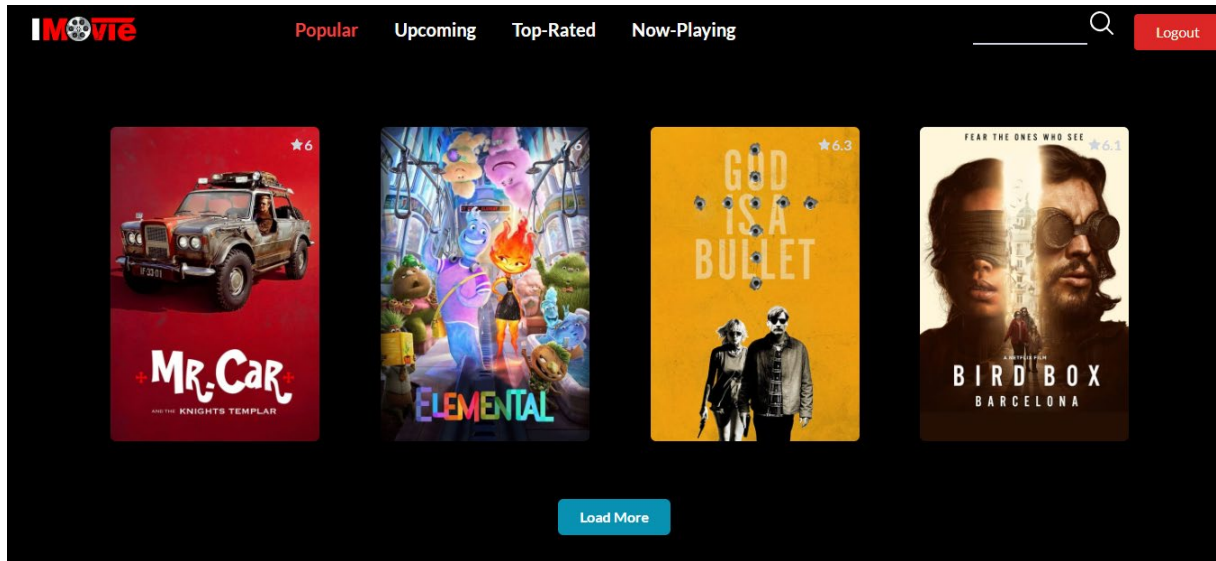
(រូបភាពទី ១៧ ៖ បង្ហាញពីខ្សែភាពយន្តដែលនឹងចាក់ផ្សាយឆាប់ៗខាងមុខ)



(រូបភាពទី ១៨ ៖ បង្ហាញពីខ្សែភាពយន្តដែលកំពុងចាក់ផ្សាយ)

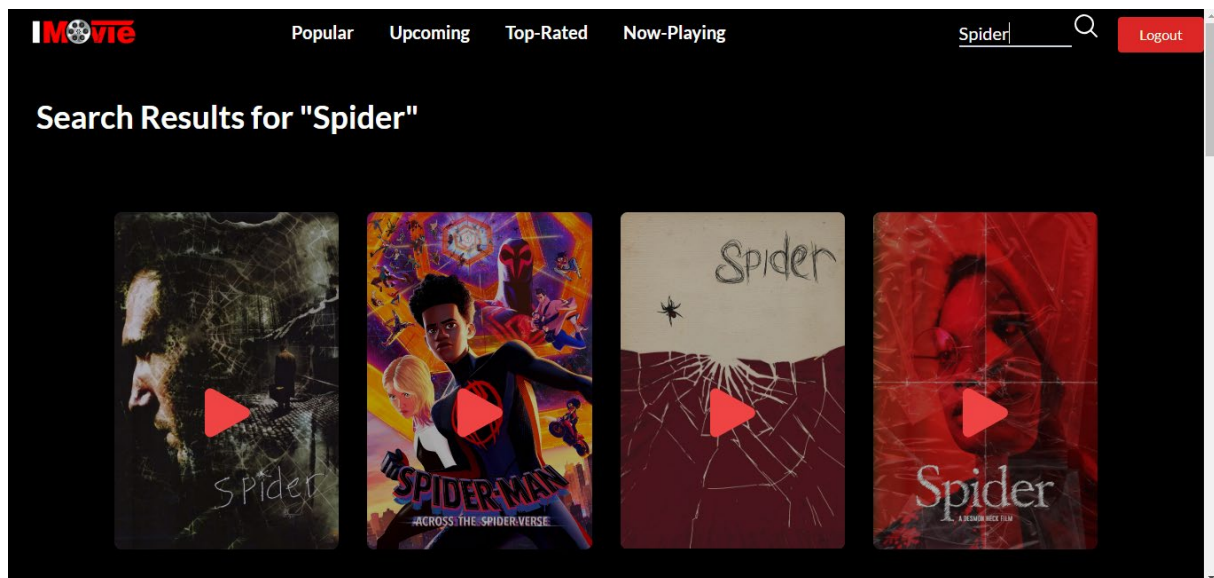
- ប៊ូតុង Load More

មានប៊ូតុង load more ដែលអាចអនុញ្ញាតឱ្យអ្នកប្រើប្រាស់ចុចមើលភាពយន្តបន្ថែមទៀតបាន ដែលប៊ូតុងមួយនេះមានគ្រប់នៅក្នុងប្រភេទសាច់រឿង។



(រូបភាពទី ១៩ ៖ បង្ហាញពីប៊ូតុង Load more)

- ស្វែងរកភាពយន្ត



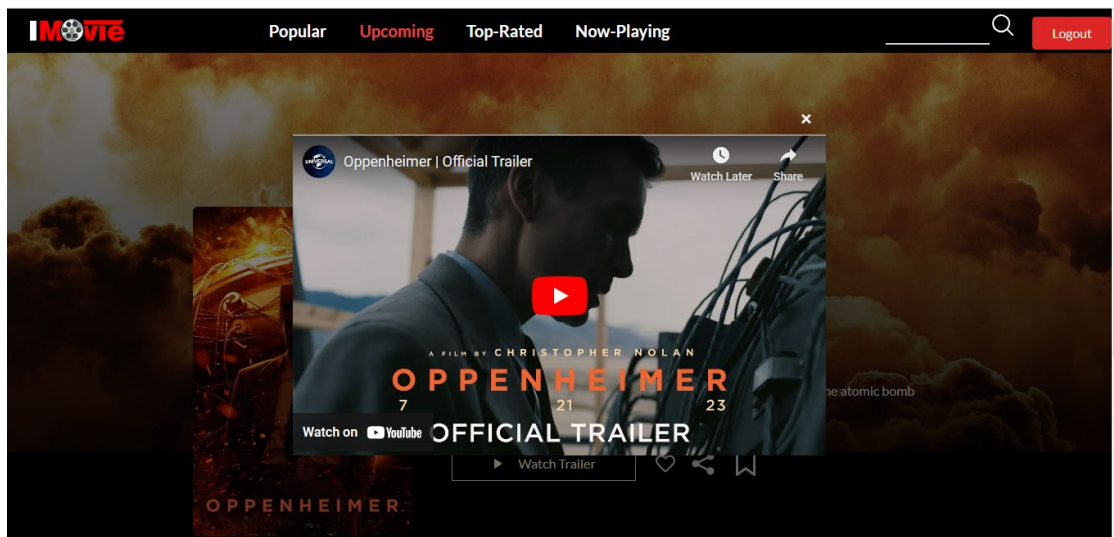
(រូបភាពទី ២០ ៖ បង្ហាញពីប៊ូតុង Search)

- ព័ត៌មានលម្អិតនៅក្នុងប្រភេទសាច់រឿងនីមួយៗ

នៅពេលដែលចុចនៅលើភាពយន្តនឹងមានការបង្ហាញជា ចំណងជើង មាតិកានៃភាពយន្ត ការទស្សនា វីដេអូខ្លីៗភាពយន្ត ការដាក់ពិន្ទុពីទស្សនិកជនទៅលើសាច់រឿង។

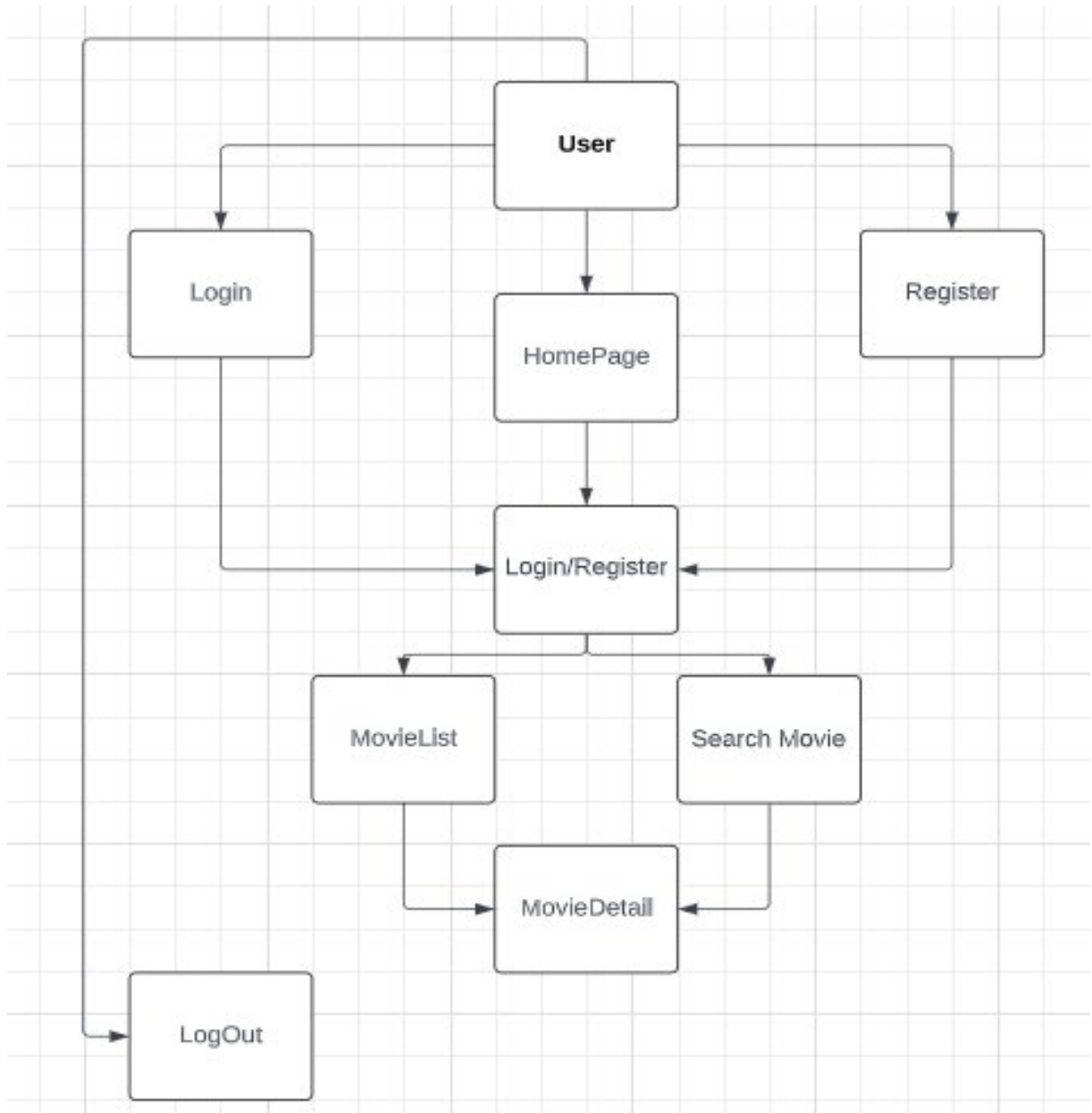


(រូបភាពទី ២១ ៖ បង្ហាញពីផ្នែកខាងក្នុងនៃខ្សែភាពយន្តនីមួយៗ)



(រូបភាពទី ២២ ៖ បង្ហាញពីផ្ទាំងសម្រាប់មើល trailer នៃខ្សែភាពយន្ត)

២.៣. ដំណើរការគ្រោងបែបសាយ



(ដ្យាក្រាម ទី ០១ ៖ បង្ហាញអំពីរចនាសម្ព័ន្ធនៃគេហទំព័រ IMOVIE)

សេចក្តីសន្និដ្ឋាន

នៅក្នុងការអភិវឌ្ឍន៍គេហទំព័រសព្វថ្ងៃ ការរចនាទំព័រមួយៗឱ្យល្អគឺចាំបាច់ណាស់សម្រាប់គេហទំព័រ ព្រោះថាបើសិនជាគេហទំព័រមួយមានការរចនាបញ្ចេញប្រព័ន្ធមិនស្អាតក៏អាចជាកត្តាមួយធ្វើឱ្យបាត់បង់ភាពទាក់ទាញពីអ្នកចូលទស្សនានិងបាត់បង់ចំណូលសម្រាប់គេហទំព័រមួយនោះផងដែរ។ ជាទូទៅប្លង់ទំព័រដែលប្រកបដោយភាពទាក់ទាញគឺធ្វើឡើងដោយបំពេញជាតុជាមូលដ្ឋាននៃការរចនាគេហទំព័រ ដោយធាតុទាំងនោះរួមមានដូចជាភាពផ្ទុយគ្នានៃពណ៌ ការរៀបចំអត្ថបទ ការជ្រើសរើសប្រភេទអក្សរ រចនាបទនៃទំព័រ ទំហំទំព័រ ក្រាហ្វិក រូបភាពដែលបានប្រើ និងភាពស្ថិតស្ថេរនៃគេហទំព័រ។ ដើម្បីបង្កើតគេហទំព័រដែលប្រកបទៅដោយស្តង់ដារក្នុងការទាក់ទាញអ្នកចូលមកទស្សនាជាក់លាក់ អ្នកអភិវឌ្ឍន៍ត្រូវតែចេះរៀបចំនិងវិភាគស្ថិតិរបស់អ្នកប្រើប្រាស់គេហទំព័រ។ ទោះបីជាមានផលលំបាកក្នុងការបង្កើតការរចនាដែលសាកសមនឹងអ្នកប្រើប្រាស់ទាំងអស់ក៏ដោយប៉ុន្តែវាអាចនឹងជាការរចនាដ៏សមរម្យក៏ដូចជាងាយស្រួលប្រើសម្រាប់អ្នកទស្សនាទូទៅ។ កាលណាការរចនាគេហទំព័រកាន់តែល្អ ពេលនោះហើយនឹងជួយនាំអ្នកទស្សនាមកកាន់គេហទំព័រកាន់តែច្រើននិងជួយឱ្យអាជីវកម្មកាន់តែមានភាពរីកចម្រើន។ ក្រោយពីបានបកស្រាយទៅលើប្រធានបទស្តីពី ការរចនាគេហទំព័រ នេះរួចមកដោយឆ្លងកាត់ទៅលើកាតិកាគន្លងន័យ និងការសរសេរកូដកន្លងមកនេះពួកយើងសម្រេចបាននូវសារណាមួយដែលអាចជួយណែនាំទៅដល់និស្សិតក៏ដូចជាអ្នកដែលមានចំណាប់អារម្មណ៍ទៅលើការបង្កើតគេហទំព័រនូវរបៀប និងវិធីក្នុងការសរសេរគេហទំព័រអោយត្រូវទៅតាមស្តង់ដារ។ ទោះបីជាមានភាពជោគជ័យទៅលើការបង្កើតគេហទំព័រដែលអាច demo ជាខ្សែភាពយន្តខ្លីៗបានក៏ដោយក៏ពួកយើងនៅតែបើកចិត្តទទួលយកនូវការស្ថាបនាផ្សេងៗពីលោកគ្រូ អ្នកគ្រូ សាស្ត្រាចារ្យ និងពីសំណាក់មិត្តនិស្សិតទាំងអស់។ ដោយសារកត្តាពេលវេលាមានកំណត់ទៅលើការប្រគល់សារណា ដូច្នេះគេហទំព័រមួយនេះនៅមានការខ្វះខាតតិចតួចទៅលើការរចនាផងដែរ។

ជាចុងក្រោយតាមរយៈមេរៀននិងវេបសាយដែលក្រុមយើងខ្ញុំបានសរសេរនិងបង្កើតឡើងនេះ យើងសង្ឃឹមថា អាចផ្តល់ជាប្រយោជន៍ក៏ដូចជាគំរូមួយទៅដល់និស្សិតនិស្សាសិស្ស និស្សិត និងអ្នកអភិវឌ្ឍន៍គេហទំព័រផ្សេងទៀតក្នុងការយកទៅសិក្សា ប្រើប្រាស់ក៏ដូចជាស្រាវជ្រាវដើម្បីបង្កើតជាវេបសាយផ្សេងៗទៀតអោយត្រូវទៅតាមស្តង់ដារ។

អនុសាសន៍

បន្ទាប់ពីការធ្វើការវិភាគទៅត្រប់ព័ត៌មានដែលបានស្រាវជ្រាវហើយនិងមតិកែលម្អពីអ្នកប្រើប្រាស់មួយចំនួន។ ក្រុមយើងខ្ញុំចង់ផ្តល់ជាអនុសាសន៍ដែលជាគោលបំណងទូទៅនៅក្នុងសារណា ដោយគ្រប់ទំព័រនីមួយៗដែលមាននៅក្នុងវេបសាយគួរតែមានរបាយការណ៍ក្នុងការរុករកដើម្បីអោយមានភាពងាយស្រួលក្នុងការប្រើប្រាស់ពីទស្សនិកជន។ មួយវិញទៀតដើម្បីបង្កើតគេហទំព័រឱ្យមានភាពស្តង់ដារគួរតែមាន CSS ដើម្បីគ្រប់គ្រងប្រភេទអក្សរទំហំប្រភេទអក្សរ និង កំណត់នូវប្លង់នៅលើគេហទំព័រទាំងមូលឱ្យមានភាពស្តង់ដារ ក៏ដូចជាដើម្បីទទួលបានរូបរាងគេហទំព័រដ៏ស្រស់ស្អាតមួយសម្រាប់អ្នកប្រើប្រាស់។ ដើម្បីឱ្យការចនាគេហទំព័រឱ្យកាន់តែមានភាពល្អប្រសើរ និងងាយស្រួលក្នុងការប្រើប្រាស់ផងដែរ។ យើងខ្ញុំសូមចែករំលែក ព្រមទាំងផ្តល់ជាអនុសាសន៍មួយចំនួនដូចខាងក្រោម៖

- **សម្រាប់សិស្សនិស្សិត** គួរធ្វើការស្វែងយល់អំពីធាតុទាំង ៤ នៃការចនាគេហទំព័រ (UX/UI, Graphic Design និង Web Development) និងសិក្សាពីមូលដ្ឋានគ្រឹះនៃភាសានីមួយៗអោយបានច្បាស់។
- **សម្រាប់អ្នកចនាគេហទំព័រ** គួរពិចារណាប្រើប្រាស់បច្ចេកទេសផ្សេងៗដើម្បីបង្កើនភាពទាក់ទាញដែលមើលឃើញ អន្តរកម្ម និងមុខងារនៃគេហទំព័ររបស់ពួកគេ។ ឧទាហរណ៍ អ្នកចនាគេហទំព័រអាចប្រើពណ៌ចម្រុះ ពុម្ពអក្សរ រូបភាព ចលនា និងសំឡេងដែលត្រូវគ្នានឹងគោលបំណងនៃគេហទំព័ររបស់ពួកគេ។
- **សម្រាប់អ្នកចនាគេហទំព័រឯករាជ្យ** គួរដកស្រង់ និងប្រមូលយកព័ត៌មានដែលជាតម្រូវការរបស់អតិថិជន ឬ clients អោយបានច្បាស់លាស់ ជៀសវាងការកែច្នៃដង ក៏ដូចជាការខកចិត្តពីសំណាក់អតិថិជន។

ក្នុងករណីចង់ដំណើរការគេហទំព័រជាលក្ខណៈអនឡាញម្ចាស់គេហទំព័រត្រូវធ្វើការទិញ Domain Name និង Hosting ជាមុនសិន ដើម្បីផ្តល់ភាពងាយស្រួលដល់អតិថិជនដែលចង់ចូលមើល និងចង់ធ្វើការជាទំនិញជាដើម។

ឯកសារយោង

<https://www.techtarget.com>

<https://edu.gcfglobal.org/en/internetbasics/what-is-the-internet/1/>

<https://www.britannica.com/technology/Internet>

<https://www.bbvaopenmind.com/en/articles/internet-changed-everyday-life/>

<https://www.theodysseyonline.com/how-internet-and-technology-changed-the-world>

<https://www.sciencenews.org/article/how-internet-will-change-world-even-more>

<https://www.usatoday.com/>

<https://www.theguardian.com/>

<https://www.scienceandmediamuseum.org.uk/objects-and-stories/short-history-internet>

<https://www.britannica.com/technology/Internet>

<https://www.coursera.org/articles/ui-design>

<https://www.nngroup.com/articles/>

<https://simplicable.com/>

<https://rapidweblaunch.com/>

<https://www.websitebuilderinsider.com/>

<https://www.exposure.com/>

<https://www.websitebuilderinsider.com/>

<https://careerfoundry.com/>

<https://www.sprakdesign.com/graphic-design-and-web-design/>

<https://dribbble.com/resources/types-of-graphic-design>

<https://www.coursera.org/articles/web-designer>

<https://edu.gcfglobal.org/en/beginning-graphic-design/layout-and-composition/1/>

<https://www.designyup.com/importance-of-layout-in-graphic-design/>

<https://99designs.com/blog/tips/typography-design/>

<https://www.masterclass.com/articles/typography-design-guide>

<https://www.gettyimages.com/detail/illustration/sport-modern-italic-alphabet-font-typography-royalty-free-illustration/1143857887>

<https://edu.gcfglobal.org/en/beginning-graphic-design/typography/1/>

ឧបសម្ព័ន្ធ

JavaScript XML , HTML, CSS, JavaScript

- apiConfig.js

```
1  const apiConfig = {
2    baseUrl: 'https://api.themoviedb.org/3/',
3    apiKey: "4113f3ad734e747a5b463cde8c55de42",
4    originalImage: (imgPath) => `https://image.tmdb.org/t/p/original/${imgPath}`,
5    w500Image: (imgPath) => `https://image.tmdb.org/t/p/w500/${imgPath}`
6  }
7
8  export default apiConfig;
```

- axiosClient.js

```
1
2  import axios from 'axios';
3  import queryString from 'query-string';
4  import apiConfig from './apiConfig';
5  const axiosClient = axios.create({
6    baseUrl: apiConfig.baseUrl,
7    headers: {
8      'Content-Type': 'application/json'
9    },
10    paramsSerializer: params => queryString.stringify({...params, api_key: apiConfig.apiKey})
11  });
12
13  axiosClient.interceptors.request.use(async (config) => config);
14
15  axiosClient.interceptors.response.use((response) => {
16    if (response && response.data) {
17      return response.data;
18    }
19    return response;
20  }, (error) => {
21    throw error;
22  });
23
24  export default axiosClient;
```

- AuthContext.js

```
1 import { createContext,useContext,useEffect,useState } from "react";
2 import { createUserWithEmailAndPassword,signInWithEmailAndPassword,signOut,onAuthStateChanged} from 'firebase/auth'
3 import {auth,db} from "../firebase.js"
4 import {setDoc,doc} from "firebase/firestore"
5
6 const AuthContext = createContext();
7
8 export function AuthContextProvider ({children}){
9   const [user,setUser]= useState({})
10
11   function signUp(email,password){
12     createUserWithEmailAndPassword(auth,email,password)
13     setDoc(doc(db,'users',email),{
14       savedShows: []
15     })
16   }
17
18   function login (email,password){
19     return signInWithEmailAndPassword(auth,email,password)
20   }
21
22   function logout(){
23     signOut(auth)
24   }
25
26   useEffect(()=>{
27     const unsubscribe =onAuthStateChanged(auth, (currentUser)=>{
28       setUser(currentUser)
29     })
30     return ()=>{
31       unsubscribe();
32     }
33   })
34   return (
35     <AuthContext.Provider value={{signUp,login,logout,user}}>
36       {children}
37     </AuthContext.Provider>
38   )
39 }
40
41 export function UserAuth (){
42   return useContext(AuthContext)
43 }
```

- Home.jsx

```
1 import requests from "../Requests/Request";
2 import Main from "../component/Main";
3 import Row from "../component/Row";
4
5 const Home = () => {
6   return(
7     <div>
8       <Main genre="popular"/>
9       <Row title="Popular" fetchURL={requests.requestPopular} rowID='1' genre="popular" />
10      <Row title="Upcoming" fetchURL={requests.requestUpcoming} rowID='2' genre="upcoming" />
11      <Row title="Trending" fetchURL={requests.requestTrending} rowID='3' genre="popular" />
12      <Row title="Now-playing" fetchURL={requests.requestNowPlaying} rowID='4' genre="now_playing" />
13      <Row title="Toprated" fetchURL={requests.requestTopRated} rowID='5' genre="top_rated" />
14    </div>
15  )
16 }
17 export default Home;
```

- Card.jsx

```
1 import { PlayIcon } from "@heroicons/react/24/solid";
2 import { Link } from "react-router-dom";
3
4 function Card({ movie }) {
5   return (
6     <Link
7       className="rounded-lg overflow-hidden relative"
8       to={` /details/${movie.id}`}
9     >
10      <img
11        src={`https://image.tmdb.org/t/p/original/${movie.poster_path}`}
12        alt={movie.original_title}
13        className="w-full object-cover"
14      />
15      <div className="dark-ov"></div>
16      <PlayIcon className="play-ic" />
17    </Link>
18  );
19 }
20
21 export default Card;
```


- Login.jsx and Request.jsx

```
1 import React,{useState} from "react";
2 import { Link, useNavigate } from "react-router-dom";
3 import { UserAuth } from "../Context/AuthContext";
4
5
6 const login = () => {
7   const [email, setEmail] = useState("");
8   const [password, setPassword] = useState("");
9   const [error, setError] = useState("");
10   const { user, login } = UserAuth();
11   const navigate = useNavigate();
12
13   const handleSubmit = async (e) => {
14     e.preventDefault();
15     setError("");
16     try {
17       await login(email, password);
18       navigate("/");
19     } catch (error) {
20       console.log(error);
21       setError(error.message);
22     }
23   };
24
25   return (
26     <div
27       className="w-full h-screen"
28       
33       <div className="bg-black/50 fixed top-0 left-0 w-full h-screen"></div>
34       <div className="fixed w-full px-4 py-24 z-50">
35         <div className="max-w-[450px] h-[600px] mx-auto bg-black/75 text-white">
36           <div className="max-w-[320px] mx-auto py-16">
37             <h1 className="text-3xl font-bold">Sign In</h1>
38             <error ? <p className="p-3 bg-red-500 my-2">{error}</p> : null>
39             <form onSubmit={handleSubmit} className="w-full flex flex-col py-4">
40               <input
41                 onChange={(e) => setEmail(e.target.value)}
42                 className="p-3 my-2 bg-gray-700 rounded"
43                 type="email"
44                 placeholder="Email"
45                 autoComplete="email"
46                 required
47               />
48               <input
49                 onChange={(e) => setPassword(e.target.value)}
50                 className="p-3 my-2 bg-gray-700 rounded"
51                 type="password"
52                 placeholder="Password"
53                 autoComplete="current-password"
54                 required
55                 minLength="6"
56               />
57               <button className="bg-red-600 py-3 my-6 rounded font-bold">
58                 Sign In
59               </button>
60               <div className="flex justify-between items-center text-sm text-gray-600">
61                 <p>
62                   <input className="mr-2" type="checkbox" />
63                   Remember me
64                 </p>
65                 <p>Need Help?</p>
66               </div>
67               <p className="py-8">
68                 <span className="text-gray-600">New to IMDb?</span>{" "}
69                 <link to="/signup">Sign Up</Link>
70               </p>
71             </form>
72           </div>
73         </div>
74       </div>
75     </div>
76   );
77 };
78
79 export default login;
```

```
1 const key = "4113f3ad734e747a5b463cde8c55de42";
2
3 const requests = {
4   requestPopular: `https://api.themoviedb.org/3/movie/popular?api_key=${key}&language=en-US&page=1`,
5   requestTopRated: `https://api.themoviedb.org/3/movie/top_rated?api_key=${key}&language=en-US&page=1`,
6   requestTrending: `https://api.themoviedb.org/3/movie/popular?api_key=${key}&language=en-US&page=1`,
7   requestNowPlaying: `https://api.themoviedb.org/3/movie/now_playing?api_key=${key}&language=en-US&page=1`,
8   requestUpcoming: `https://api.themoviedb.org/3/movie/upcoming?api_key=${key}&language=en-US&page=1`,
9
10 };
11
12 export default requests;
```


- MovieList.jsx

```
1 import React, { useEffect, useState } from "react";
2 import axios from "axios";
3 import { useParams } from "react-router-dom";
4 import Movie from "../component/Movie";
5
6
7
8 const MovieList = () => {
9   const params = useParams();
10   const key = "4113f3ad734e747a5b463cde8c55de42";
11
12
13   const [movies, setMovies] = useState([]);
14   const [page, setPage] = useState(1);
15
16   useEffect(() => {
17     fetchData(page);
18   }, [page]);
19
20
21   const fetchData = async (pageNum) => {
22     axios
23       .get(
24         `https://api.themoviedb.org/3/movie/${params.genre}?api_key=${key}&language=en-US&page=${pageNum}`
25       )
26       .then((response) => {
27         setMovies((prevstate) => [...prevstate, ...response.data.results]);
28       })
29       .catch((error) => {
30         console.log(error);
31       });
32   };
33
34
35   console.log(movies)
36
37   const loadMore = () => {
38     setPage((prevstate) => prevstate + 1);
39   };
40
41   return (
42     <div>
43
44       <div className="mx-auto py-10 px-6 max-w-[90%]">
45         <div className="grid grid-cols-2 gap-y-10 gap-x-6 md:grid-cols-2 lg:grid-cols-3 xl:grid-cols-4 2xl:grid-cols-5 xl:gap-x-8 mt-12">
46           {movies.map((item) => (
47             <Movie item={item} ></Movie>
48           ))}
49         </div>
50         <div className="flex items-center justify-center mt-10">
51           <button className="bg-cyan-600 hover:bg-cyan-800 text-white font-bold py-2 px-4 rounded " onClick={loadMore} disabled={page>15 ? true: false}>
52             Load More
53           </button>
54         </div>
55       </div>
56     </div>
57   );
58 };
59
60
61 export default MovieList;
62
```

- NowPlay.jsx

```
1 import React, { useEffect, useState } from "react";
2 import axios from "axios";
3
4 import Movie from "../component/Movie";
5 const NowPlay = () => {
6
7   const key = "4113f3ad734e747a5b463cde8c55de42";
8
9
10  const [movies, setMovies] = useState([]);
11  const [page, setPage] = useState(1);
12
13  useEffect(() => {
14    fetchData(page);
15  }, [page]);
16
17
18  const fetchData = async (pageNum) => {
19    axios
20      .get(
21        `https://api.themoviedb.org/3/movie/now_playing?api_key=${key}&language=en-US&page=${pageNum}`
22      )
23      .then((response) => {
24        setMovies((prevstate) => [...prevstate, ...response.data.results]);
25      })
26      .catch((error) => {
27        console.log(error);
28      });
29  };
30
31
32  console.log(movies)
33
34  const loadMore = () => {
35    setPage((prevstate) => prevstate + 1);
36  };
37
38  return (
39
40
41    <div className="rows">
42      <h1 className="movie">Now-Playing Movies</h1>
43      <div className="mx-auto py-10 px-6 max-w-[90%]">
44
45        <div className="grid grid-cols-2 gap-y-10 gap-x-6 md:grid-cols-2 lg:grid-cols-3 xl:grid-cols-4 2xl:grid-cols-5 xl:gap-x-8 mt-12">
46          {movies.map((item) => (
47            <Movie item={item} ></Movie>
48          ))}
49        </div>
50        <div className="flex items-center justify-center mt-10">
51
52          <button className="bg-cyan-600 hover:bg-cyan-800 text-white font-bold py-2 px-4 rounded " onClick={loadMore} disabled={page>15 ? true: false}>
53            Load More
54          </button>
55
56        </div>
57      </div>
58    </div>
59
60  );
61  };
62
63  export default NowPlay;
64
```


- MovieDetails.jsx and Navbar.jsx

[illegible]

```

1 import { MagnifyingGlassIcon } from "@heroicons/react/24/solid";
2 import { useState } from "react";
3 import { NavLink, useNavigate } from "react-router-dom";
4 import { UserAuth } from "../Context/AuthContext";
5
6 function Nav() {
7   const { user, logout } = UserAuth();
8
9   const [searchKey, setSearchKey] = useState("");
10
11   const navigate = useNavigate();
12
13   const search = (e) => {
14     e.preventDefault();
15     navigate(`/search/${searchKey}`);
16   };
17
18   const handleLogout = async () => {
19     try {
20       await logout();
21       navigate("/");
22     } catch (error) {
23       console.log(error);
24     }
25   };
26
27   return (
28     <nav className="navbar sticky-top">
29       <NavLink to="/">
30         <div id="title">
31           
32         </div>
33       </NavLink>
34       {user?.email ? (
35         <div className="listbar">
36           <NavLink
37             className={({ isActive }) =>
38               isActive ? "active-text" : "non-active-text"
39             }
40             to={"/popular"}
41           >
42             Popular
43           </NavLink>
44           <NavLink
45             className={({ isActive }) =>
46               isActive ? "active-text" : "non-active-text"
47             }
48             to={"/upcoming"}
49           >
50             Upcoming
51           </NavLink>
52           <NavLink
53             className={({ isActive }) =>
54               isActive ? "active-text" : "non-active-text"
55             }
56             to={"/top_rated"}
57           >
58             Top-Rated
59           </NavLink>
60           <NavLink
61             className={({ isActive }) =>
62               isActive ? "active-text" : "non-active-text"
63             }
64             to={"/now_playing"}
65           >
66             Now-Playing
67           </NavLink>
68         </div>
69       ) : (
70         <div className="search">
71           <form onSubmit={search}>
72             <input
73               type="text"
74               className="input"
75               onChange={(e) => setSearchKey(e.target.value)}
76             />
77             <button type="submit" className="Search">
78               <MagnifyingGlassIcon className="w-8 h-8" />
79             </button>
80
81             <button
82               onClick={handleLogout}
83               className="logout"
84             >
85               Logout
86             </button>
87
88             </form>
89           </div>
90         </div>
91       ) : (
92         <div>
93           <NavLink to="/signIn">
94             <button className="text-[#FFFE3] pr-4">Sign In</button>
95           </NavLink>
96           <NavLink to="/signUp">
97             <button className="text-[#FFFE3] px-6 py-2 rounded cursor-pointer bg:red-600">
98               Sign Up
99             </button>
100           </NavLink>
101         </div>
102       )
103     </nav>
104   );
105 }
106
107 export default Nav;

```

- ProtectedRoute.jsx



```
1  import React from 'react'
2  import {Navigate} from "react-router-dom"
3  import { UserAuth } from '../Context/AuthContext'
4
5  const ProtectedRoute = ({children}) => {
6      const {user} = UserAuth()
7      if(!user){
8          return <Navigate to="/"></Navigate>
9      }else{
10         return children
11     }
12
13 }
14
15 export default ProtectedRoute
```

- Searched.jsx

```

1 import { useEffect, useState } from "react";
2 import { useParams } from "react-router-dom";
3
4
5 import Card from "../Card";
6
7
8 function Searched() {
9   const { title } = useParams();
10  const [searchedMovies, setSearchedMovies] = useState([]);
11
12  useEffect(() => {
13    getSearchMovie();
14  }, [title]);
15
16  const getSearchMovie = async () => {
17    const response = await fetch(
18      `https://api.themoviedb.org/3/search/movie?api_key=4113f3ad734e747a5b463cde8c55de42&language=en-US&query=${title}&page=1&include_adult=false`
19    );
20    const data = await response.json();
21    setSearchedMovies(data.results);
22  };
23
24  return (
25
26    <section className="px-8 bg-black">
27      <h1 className="text-4xl font-bold text-mono text-white pt-5">
28        Search Results for "{title}"
29      </h1>
30      <div className="mx-auto py-10 px-6 max-w-[90%]">
31        <div className="grid grid-cols-2 gap-y-10 gap-x-6 md:grid-cols-2 lg:grid-cols-3 xl:grid-cols-4 2xl:grid-cols-5 xl:gap-x-8 mt-12">
32          {searchedMovies.length &&
33            searchedMovies.map((movie) => {
34              return <Card key={movie.id} movie={movie} />;
35            })}
36        </div>
37      </div>
38    </section>
39
40  );
41 }
42
43 export default Searched;

```

- Row.jsx

```

1 import axios from "axios";
2 import { useEffect, useState } from "react"
3 import { useNavigate } from "react-router-dom";
4 import Movie from "../Movie";
5 import { FiChevrnsRight } from 'react-icons/fi';
6
7
8 const Row = ({ title, fetchURL, rowID, genre }) => {
9   const [movies, setMovies] = useState([]);
10   const navigate = useNavigate();
11
12   useEffect(() => {
13     axios.get(fetchURL).then((response) => {
14       setMovies(response.data.results);
15       console.log(response.data.results);
16     })
17   }, [fetchURL]);
18
19   const slideLeft = () => {
20     var slider = document.getElementById('slider' + rowID)
21     slider.scrollLeft = slider.scrollLeft - 500;
22   };
23
24   const slideRight = () => {
25     var slider = document.getElementById('slider' + rowID)
26     slider.scrollRight = slider.scrollRight + 500;
27   };
28
29   const handleClick = () =>{
30     navigate(`/${genre}`)
31   }
32   return(
33     <div className="rows">
34       <div className="flex flex-row items-center">
35         <h2 className="text-[#FFFDDE3] font-bold md:text-xl p-4 cursor-pointer">{title}</h2>
36         {genre ? <FiChevrnsRight onClick={handleClick} className="text-white cursor-pointer"/> : null}
37       </div>
38       <div className="relative flex items-center ml-2 group">
39         <div
40           className="slideleft"
41           size={30}
42           onClick={slideLeft}
43         />
44         <div id={'slider'+ rowID} className='slide'>
45           {movies.map((item) => {
46             return (
47               <Movie item={item} genre={genre}></Movie>
48             );
49           })}
50         </div>
51         <div
52           className='slideright'
53           size={30}
54           onClick={slideRight}
55         />
56       </div>
57     </div>
58   )
59 }
60 export default Row;

```

- SignUp.jsx

```

1 import React, { useState } from "react";
2 import { Link, useNavigate } from "react-router-dom";
3
4 import { UserAuth } from "../Context/AuthContext";
5
6 const SignUp = () => {
7   const [email, setEmail] = useState("");
8   const [password, setPassword] = useState("");
9
10  const { user, signUp } = UserAuth();
11  const navigate = useNavigate();
12
13  const handleSubmit = async (e) => {
14    e.preventDefault();
15
16    try {
17      await signUp(email, password);
18      navigate("/");
19    } catch (error) {
20      console.log(error);
21    }
22  };
23
24  return (
25    <div>
26      <div className="w-full h-screen">
27        
30        <div className="bg-black/60 fixed top-0 left-0 w-full h-screen"></div>
31        <div className="fixed w-full px-4 py-24 z-50">
32          <div className="max-w-[450px] h-[600px] mx-auto bg-black/75 text-white">
33            <div className="max-w-[320px] mx-auto py-16">
34              <h1 className="text-3xl font-bold">Sign Up</h1>
35              <form>
36                <input
37                  className="w-full flex flex-col py-4"
38                  type="email"
39                  placeholder="Email"
40                  autoComplete="email"
41                  onChange={(e) => setEmail(e.target.value)}
42                  required
43                />
44                <input
45                  className="p-3 my-2 bg-gray-700 rounded"
46                  type="password"
47                  placeholder="Password"
48                  autoComplete="current-password"
49                  onChange={(e) => setPassword(e.target.value)}
50                  required
51                  minLength="6"
52                />
53                <button className="bg-red-600 py-3 my-6 rounded font-bold">
54                  Sign Up
55                </button>
56                <div className="flex justify-between items-center text-sm text-gray-600">
57                  <p>
58                    <input className="mr-2" type="checkbox" />
59                    Remember me
60                  </p>
61                  <p>Need Help?</p>
62                </div>
63                <p className="py-8">
64                  <span className="text-gray-600">
65                    Already subscribed to IMovie?
66                  </span>[" "]
67                  <Link to="/signin">Sign In</Link>
68                </p>
69              </form>
70            </div>
71          </div>
72        </div>
73      </div>
74    </div>
75  );
76
77 export default SignUp;

```

- TopRated.jsx

```
1 import React, { useEffect, useState } from "react";
2 import axios from "axios";
3
4 import Movie from "../component/Movie";
5 const TopRated = () => {
6
7   const key = "4113f3ad734e747a5b463cde8c55de42";
8
9
10  const [movies, setMovies] = useState([]);
11  const [page, setPage] = useState(1);
12
13  useEffect(() => {
14    fetchData(page);
15  }, [page]);
16
17
18  const fetchData = async (pageNum) => {
19    axios
20      .get(
21        `https://api.themoviedb.org/3/movie/top_rated?api_key=${key}&language=en-US&page=${pageNum}`
22      )
23      .then((response) => {
24        setMovies((prevstate) => [...prevstate, ...response.data.results]);
25      })
26      .catch((error) => {
27        console.log(error);
28      });
29  };
30  console.log(movies)
31
32  const loadMore = () => {
33    setPage((prevstate) => prevstate + 1);
34  };
35
36  return (
37
38
39    <div className="rows">
40      <h1 className="movie">Top-Rated Movies</h1>
41      <div className="mx-auto py-10 px-6 max-w-[90%]">
42
43        <div className="grid grid-cols-2 gap-y-10 gap-x-6 md:grid-cols-2 lg:grid-cols-3 xl:grid-cols-4 2xl:grid-cols-5 xl:gap-x-8 mt-12">
44          {movies.map((item) => (
45            <Movie item={item} ></Movie>
46          ))}
47        </div>
48        <div className="flex items-center justify-center mt-10">
49
50          <button className="bg-cyan-600 hover:bg-cyan-800 text-white font-bold py-2 px-4 rounded " onClick={loadMore} disabled={page>15 ? true: false}>
51            Load More
52          </button>
53
54        </div>
55      </div>
56    </div>
57
58  );
59  };
60
61  export default TopRated;
62
```

- Upcoming.jsx

```
1 import React, { useEffect, useState } from "react";
2 import axios from "axios";
3
4 import Movie from "../component/Movie";
5 const Upcoming = () => {
6
7   const key = "4113f3ad734e747a5b463cde8c55de42";
8
9
10  const [movies, setMovies] = useState([]);
11  const [page, setPage] = useState(1);
12
13  useEffect(() => {
14    fetchData(page);
15  }, [page]);
16
17
18  const fetchData = async (pageNum) => {
19    axios
20      .get(
21        `https://api.themoviedb.org/3/movie/upcoming?api_key=${key}&language=en-US&page=${pageNum}`
22      )
23      .then((response) => {
24        setMovies((prevstate) => [...prevstate, ...response.data.results]);
25      })
26      .catch((error) => {
27        console.log(error);
28      });
29  };
30
31
32  console.log(movies)
33
34  const loadMore = () => {
35    setPage((prevstate) => prevstate + 1);
36  };
37
38  return (
39
40
41    <div className="rows">
42      <h1 className="movie">Upcoming Movies</h1>
43      <div className="mx-auto py-10 px-6 max-w-[90%]">
44
45        <div className="grid grid-cols-2 gap-y-10 gap-x-6 md:grid-cols-2 lg:grid-cols-3 xl:grid-cols-4 2xl:grid-cols-5 xl:gap-x-8 mt-12">
46          {movies.map((item) => (
47            <Movie item={item} ></Movie>
48          ))}
49        </div>
50        <div className="flex items-center justify-center mt-10">
51
52          <button className="bg-cyan-600 hover:bg-cyan-800 text-white font-bold py-2 px-4 rounded " onClick={loadMore} disabled={page>15 ? true: false}>
53            Load More
54          </button>
55
56        </div>
57      </div>
58    </div>
59
60  );
61  };
62
63  export default Upcoming;
64
```

- Index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
6     <meta name="viewport" content="width=device-width, initial-scale=1" />
7     <meta name="theme-color" content="#000000" />
8     <meta
9       name="description"
10      content="Web site created using create-react-app"
11    />
12    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
13    <!--
14      Manifest.json provides metadata used when your web app is installed on a
15      user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/
16    -->
17    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
18    <!--
19      Notice the use of %PUBLIC_URL% in the tags above.
20      It will be replaced with the URL of the 'public' folder during the build.
21      Only files inside the 'public' folder can be referenced from the HTML.
22
23      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
24      work correctly both with client-side routing and a non-root public URL.
25      Learn how to configure a non-root public URL by running 'npm run build'.
26    -->
27    <title>React App</title>
28    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/twitter-bootstrap/4.1.3/css/bootstrap.min.css">
29    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/ionicons/2.0.1/css/ionicons.min.css">
30    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KK94CHFLLe+Ny2mCwGh91rCG6gU4hK52Hd+Ye+H/SXH381u52Ly+H9N+20Z" crossorigin="anonymous">
31  </head>
32  <body>
33    <noscript>You need to enable JavaScript to run this app.</noscript>
34    <div id="root"></div>
35    <!--
36      This HTML file is a template.
37      If you open it directly in the browser, you will see an empty page.
38
39      You can add webfonts, meta tags, or analytics to this file.
40      The build step will place the bundled scripts into the <body> tag.
41
42      To begin the development, run 'npm start' or 'yarn start'.
43      To create a production bundle, use 'npm run build' or 'yarn build'.
44    -->
45    <script src="https://www.gstatic.com/firebasejs/10.0.0/firebase-app-compat.js"></script>
46    <script src="https://www.gstatic.com/firebasejs/10.0.0/firebase-firestore-compat.js"></script>
47  </body>
48 </html>
```

- Firebase.js



```
1 // Import the functions you need from the SDKs you need
2 import { initializeApp } from "firebase/app";
3 import { getAnalytics } from "firebase/analytics";
4 import { getAuth } from "firebase/auth";
5 import { getFirestore } from "firebase/firestore";
6 // TODO: Add SDKs for Firebase products that you want to use
7 // https://firebase.google.com/docs/web/setup#available-libraries
8
9 // Your web app's Firebase configuration
10 // For Firebase JS SDK v7.20.0 and later, measurementId is optional
11 const firebaseConfig = {
12   apiKey: "AIzaSyDsp5Qjr_z8tNNxGXjvvtgdkBmWx_yxDJo",
13   authDomain: "imovie-thesis-ie4c1.firebaseio.com",
14   projectId: "imovie-thesis-ie4c1",
15   storageBucket: "imovie-thesis-ie4c1.appspot.com",
16   messagingSenderId: "654866048799",
17   appId: "1:654866048799:web:c3086ef9bd69a59f1ef22a",
18   measurementId: "G-WNYV9HDD18"
19 };
20
21 // Initialize Firebase
22 const app = initializeApp(firebaseConfig);
23 const analytics = getAnalytics(app);
24 export const auth = getAuth(app);
25 export const db = getFirestore(app);
```

- App.js

```

1  import { Route, Routes } from 'react-router-dom';
2  import Home from './Page/Home';
3
4  import Nav from './component/Navbar';
5  import { MovieDetail } from './Page/MovieDetail';
6  import Popular from './Page/Popular';
7  import Upcoming from './Page/Upcoming';
8  import NowPlay from './Page/NowPlay';
9  import TopRated from './Page/TopRated';
10 import Searched from './component/Searched';
11 import { Footer } from './Page/Footer';
12 import Login from './Page/Login';
13 import SignUp from './Page/SignUp';
14 import ProtectedRoute from './component/ProtectedRoute';
15 import { AuthContextProvider } from './Context/AuthContext';
16
17
18 function App() {
19   return (
20     <div>
21       <AuthContextProvider>
22         <Nav />
23
24         <Routes>
25
26           <Route path="/" element={<Home />}></Route>
27
28           <Route path="/popular" element={
29             <ProtectedRoute>
30               <Popular />
31             </ProtectedRoute>
32           }></Route>
33           <Route path="/upcoming" element={
34             <ProtectedRoute>
35               <Upcoming />
36             </ProtectedRoute>
37           }></Route>
38           <Route path="/now_playing" element={
39             <ProtectedRoute>
40               <NowPlay /></ProtectedRoute>
41           }></Route>
42           <Route path="/top_rated" element={
43             <ProtectedRoute>
44               <TopRated /></ProtectedRoute>
45           }></Route>
46
47           <Route path="/signIn" element={<Login />}></Route>
48           <Route path="/signUp" element={<SignUp />}></Route>
49           { /* <Route element={<Detail />} path="/:genre/:id" /> */ }
50           <Route element={<MovieDetail />} path="/:genre/:id" />
51
52           <Route
53             path="/search/:title"
54             element={
55               <ProtectedRoute>
56                 <Searched />
57               </ProtectedRoute>
58             }
59           ></Route>
60         </Routes>
61
62         <Footer />
63       </AuthContextProvider>
64     </div>
65   )
66 }
67 export default App;
68
69

```

- Index.js



```
1  import React from 'react';
2  import ReactDOM from 'react-dom/client';
3  import './index.css';
4  import App from './App';
5  import reportWebVitals from './reportWebVitals';
6  import { HashRouter } from 'react-router-dom';
7
8  const root = ReactDOM.createRoot(document.getElementById('root'));
9  root.render(
10    <HashRouter>
11      <App />
12    </HashRouter>
13  );
14  reportWebVitals();
15
```

- Index.css

[illegible]