



សាកលវិទ្យាល័យតួមិទ្តនីតិសាស្ត្រ និង វិទ្យាសាស្ត្រសេដ្ឋកិច្ច

Université Royale de Droit et des Sciences Économiques

Royal university of Law and Economics



សារណាបញ្ចប់ការសិក្សា

ការគ្រប់គ្រងប្រព័ន្ធទ្វារកំរៀនតាមបែបឌីជីថល

Digital Classroom Management System

ស្រាវជ្រាវពីថ្ងៃទី ៨ ខែ មេសា ឆ្នាំ ២០២៤ ដល់ថ្ងៃទី ១០ ខែ សីហា ឆ្នាំ ២០២៤

ស្រាវជ្រាវដោយ

និស្សិតឈ្មោះ: **កញ្ញា ផល ម៉ាឌីណា**

កញ្ញា សៀនពេជ្រអេស្ទីដា

ថ្នាក់បរិញ្ញាបត្រ **សេដ្ឋកិច្ចព័ត៌មានវិទ្យា**

ជំនាន់ទី **១៩**

ណែនាំដោយ

សាស្ត្រាចារ្យជំនួយ **បណ្ឌិត ទិន ហេង**

ឆ្នាំចូលសិក្សា **២០២០**

ឆ្នាំសរសេរសារណា **២០២៤**

សេចក្តីថ្លែងអំណរគុណ

យើងខ្ញុំឈ្មោះ ផល ម៉ាឌីណា និង សៀន ពេជ្រអល្លីដា ជានិស្សិតនៃមហាវិទ្យាល័យ សេដ្ឋកិច្ចព័ត៌មាន វិទ្យា ថ្នាក់បរិញ្ញាបត្រ សេដ្ឋកិច្ចព័ត៌មានវិទ្យា ជំនាន់ទី១៩ ក្នុងឆ្នាំសិក្សា ២០២០-២០២៤ នៃសកលវិទ្យាល័យ ភូមិន្ទនីតិសាស្ត្រ និងវិទ្យាសាស្ត្រសេដ្ឋកិច្ច ។

យើងខ្ញុំទាំងពីរនាក់សូមធ្វើការអរគុណ ដឹងគុណកតញ្ញូ និងសេចក្តីគោរពជូនដល់លោកអ្នកគុណ ទាំងឡាយមាន៖

លោកឪពុក អ្នកម្តាយដែលបានផ្តល់កំណើតឱ្យកូនៗ បានរស់រានមានជីវិត និងខិតខំចិញ្ចឹមបីបាច់ថែរក្សា មើលការខុសត្រូវណែនាំទំនុកបម្រុងផ្តល់ឱ្យនូវអ្វីដែលចាំបាច់ក្នុងការអប់រំកូនឱ្យក្លាយទៅជាមនុស្សម្នាក់ដែល មានសក្តានុពល អំណត់អត់ធ្មត់នៅក្នុងគ្រួសារ និងសង្គមជាតិទាំងមូលផងដែរ។

ឯកឧត្តមបណ្ឌិតសាកលវិទ្យាធិការ សាកលវិទ្យាធិការរង ព្រឹទ្ធបុរស ព្រឹទ្ធបុរសរង នៃសាកលវិទ្យាល័យ ភូមិន្ទនីតិសាស្ត្រ និងវិទ្យាសាស្ត្រសេដ្ឋកិច្ចដែលតែងតែស្វែងរកនូវភាពងាយស្រួលដល់ការសិក្សាជូនដល់និស្សិត គ្រប់រូបរួមទាំងខ្ញុំទាំងពីរនាក់ ដើម្បីឱ្យការសិក្សាមានភាពរលូនទៅមុខប្រកបដោយប្រសិទ្ធភាព នាក់ឡុងពេល សិក្សាថ្នាក់បរិញ្ញាបត្រសេដ្ឋកិច្ចព័ត៌មានវិទ្យានេះ។

លោក លោកស្រីបុគ្គលិកសិក្សា និងលោកគ្រូ អ្នកគ្រូសាស្ត្រាចារ្យទាំងអស់ដែលតែងតែ ខិតខំបង្ហាត់ បង្រៀន ស្រាវជ្រាវនូវ មេរៀនថ្មីៗ ចែករំលែករាល់បទពិសោធន៍ផ្សេងៗ ដល់និស្សិតដោយក្តីអាណិតស្រឡាញ់ ចេញពីបេះដូងចាប់តាំងពីថ្នាក់ឆ្នាំមូលដ្ឋាន ដល់បញ្ចប់ថ្នាក់បរិញ្ញាបត្រនេះ។

សាស្ត្រាចារ្យជំនួយបណ្ឌិត **ទិន ហេង** ដែលបានណែនាំក្នុងការសរសេរសារណាបញ្ចប់ថ្នាក់បរិញ្ញាបត្រ សេដ្ឋកិច្ចព័ត៌មានវិទ្យានេះ។ លោកបានលះបង់នូវពេលវេលាដ៏មានតម្លៃរបស់លោកមកបង្ហាត់បង្រៀនស្ទុះស្ទែង រកនូវឯកសារពាក់ព័ន្ធនានាត្រួតពិនិត្យកែសម្រួលពិភាក្សាផ្តល់នូវដំបូន្មាន ឬគំនិតក្នុងការសរសេរសារណាបញ្ចប់ ថ្នាក់បរិញ្ញាបត្រនេះ។ ដោយមានការជួយជ្រោមជ្រែងពីសំណាក់លោកសាស្ត្រាចារ្យអស់ពីចិត្ត ទើបធ្វើឱ្យ សារ ណានេះ បញ្ចប់ដោយជោគជ័យជាស្ថាពរយ៉ាងរលូន។

ជាចុងក្រោយ យើងខ្ញុំទាំងពីររូបសូមគោរពជូនពរដល់អ្នកមានគុណទាំងពីរ ឯកឧត្តមបណ្ឌិតសាកលវិទ្យា ធិការ សាកលវិទ្យាធិការរង ព្រឹទ្ធបុរស ព្រឹទ្ធបុរសរង សាស្ត្រាចារ្យគ្រប់មុខវិជ្ជាទាំងអស់ ព្រមទាំងនិស្សិតគ្រប់ រូប សូមទទួលបាននូវពុទ្ធពរទាំង ៤ប្រការគឺ អាយុ វណ្ណៈ សុខៈ ពលៈ កុំបីឃ្លៀងឃ្លាតឡើយ ព្រមទាំងចៀស ចាក ផុត ពីជំងឺទាំងពួង។

អារម្ភកថា

នៅក្នុងយុគសម័យឌីជីថលនេះ បច្ចេកវិទ្យាត្រូវបានគ្របដណ្តប់លើគ្រប់វិស័យទៅហើយមិនថាកសិកម្ម ឧស្សាហកម្មទូរគមនាគមន៍ ប្រព័ន្ធផ្សព្វផ្សាយ ការកំសាន្ត ពាណិជ្ជកម្ម វិស័យអប់រំ។ល។ វាបានជួយសម្រួល ដល់ ការងាររបស់មនុស្សជាតិបានមួយផ្នែកធំរួមទាំងជីវភាពរស់នៅផងដែរ។ កំណើននៃការកើនឡើងចំនួនមនុស្សនៅ លើពិភពលោកវាជាបញ្ហាមួយនៃការកើតឡើងនូវតម្រូវការជាច្រើនឥតឈប់ឈរស្របជាមួយ និងការ គាំទ្រនៅ បញ្ហានេះទើបអ្នកស្រាវជ្រាវជាច្រើនរូបបានខិតខំបង្កើតនូវគំនិតប្រឌិតថ្មីៗជាច្រើនទៅលើបច្ចេកវិទ្យាដើម្បីបំពេញ សេចក្តីត្រូវការ របស់មនុស្សជាតិនៅលើពិភពលោកនេះដែលវាជាហេតុធ្វើឱ្យបច្ចេកវិទ្យាកាន់តែវិវត្ត ទំនើបទៅៗ ពី មួយដំណាក់កាលទៅមួយដំណាក់កាលយ៉ាងឆាប់រហ័សទើបធ្វើឱ្យមានការនិយាយតៗគ្នាថា បច្ចេកវិទ្យានៅ សម័យសតវត្សទី២១នេះ បានកែប្រែរបៀបរស់នៅរបស់មនុស្សជាតិស្ទើរតែទាំងស្រុង។

បើក្រឡេកទៅមើល ក្នុងទិដ្ឋភាពសង្គមផ្ទាល់យើងប្រាកដជាឃើញនូវភាពខ្វះចន្លោះមួយចំនួនដែលមិន ទាន់បានបំពេញនៅឡើយ។ ដូចនេះហើយទើបពួកខ្ញុំទាំងពីរនាក់បានធ្វើការសម្រេចចិត្តស្រាវជ្រាវទៅលើផ្នែក គេ ហទំព័រ ដែលសិក្សាផ្តោតសំខាន់ទៅលើការគ្រប់គ្រងទិន្នន័យថ្នាក់រៀនតាមបែបឌីជីថល។ហើយពួកខ្ញុំទាំងពីរ នាក់សង្ឃឹមថាការសិក្សាស្រាវជ្រាវនូវប្រធានបទមួយនេះ បានចូលរួមចំណែកក្នុងការអភិវឌ្ឍន៍វិស័យបច្ចេកវិទ្យា បានមួយកម្រិតជាពិសេសលើផ្នែកគេហទំព័រនេះ។

រាល់សកម្មភាព ឬការសិក្សាស្រាវជ្រាវមួយដែលមានលទ្ធផលល្អប្រកបដោយប្រសិទ្ធភាពទៀតនោះ តែង តែមាននូវរាល់ឧបសគ្គឆ្លងកាត់ជាច្រើន។ ដូចគ្នានេះដែរការយកប្រធានបទនេះមកធ្វើការស្រាវជ្រាវចងក្រង ធ្វើជា សារណា ពួកយើងទាំងពីរនាក់បានជួបបញ្ហាជាច្រើន ដូចជាការខ្វះខាតនូវឯកសារ ចំណេះដឹងមួយចំនួន ។ល។ជាសំណាងល្អដោយទទួលបានការណែនាំពីសាស្ត្រាចារ្យជំនួយបណ្ឌិត ទិន ហេង និងការស្វែងរកនូវ ទិន្នន័យពាក់ព័ន្ធនានាតាមប្រព័ន្ធអ៊ីនធឺណិត (Internet) ទើបធ្វើឱ្យសារណាមួយនេះបញ្ចប់ដោយ ជោគជ័យ។ ទោះបីពួកយើងទាំងពីរនាក់ខិតខំអស់ពីសមត្ថភាពយ៉ាងណាក្តីក៏ការសិក្សាស្រាវជ្រាវមួយនេះនៅ មានចន្លោះខ្វះ ខាត បញ្ហាបន្តិចបន្តួចជាថាហេតុ អាស្រ័យហេតុនេះពួកខ្ញុំទាំងពីរអ្នកសង្ឃឹមថាអ្នកអាននឹង អធ្យាស្រ័យអត់ឱ នឱ្យពួកយើងទាំងពីរនាក់ រាល់កំហុសដែលកើតមានឡើង ទាំងចេតនា នឹងអចេតនាដោយក្តី អនុគ្រោះ។

មាតិកា

ទំព័រ

សេចក្តីផ្តើម.....	1
១. លំនាំបញ្ហានៃការស្រាវជ្រាវ	1
២. ចំណោទបញ្ហា.....	1
៣. គោលបំណង.....	2
៤. វិសាលភាពនិងដែនកំណត់សិក្សា	2
៥. សារៈសំខាន់នៃការសិក្សា	3
៦. វិធីសាស្ត្រនៃការស្រាវជ្រាវ.....	3
៧. រចនាសម្ព័ន្ធនៃការស្រាវជ្រាវ	3

ជំពូកទី១ រំលឹកទ្រឹស្តីពាក់ព័ន្ធ

១.១. អំពីគេហទំព័រ	4
១.១.១. តើអ្វីទៅជាគេហទំព័រ ?	4
១.១.២. ប្រភេទនៃគេហទំព័រ	4
១.១.៣. តើអ្វីទៅជាDigital Classroom?	6
១.១.៤. ប្រភេទនៃគេហទំព័រ Digital Classroom	6

១.១.៥.ប្រវត្តិនៃការបង្កើត Digital Classroom.....	6
១.១.៦.គោលបំណងក្នុងការបង្កើត Digital Classroom	7
១.២.ទ្រឹស្តីពាក់ព័ន្ធការបង្កើត DCMS	7
១.២.១ Tailwind CSS.....	7
១.២.២ Next JS	8
១.២.៣ JavaScript.....	8
១.២.៤ SpringBoot.....	9
១.៣ ទ្រឹស្តីពាក់ព័ន្ធនឹង Database	9
១.៣.២ អ្វីទៅជា Database Management System (DBMS)?	9
១.៣.៣ ទ្រឹស្តី System Development Life Cycle (SDLC).....	10
១.៣.៤ ទ្រឹស្តី PostgreSQL.....	12
១.៤ ឧបករណ៍និងវេបសាយជំនួយ	12
១.៤.១ ទ្រឹស្តីពាក់ព័ន្ធនឹង Microsoft	12
១.៤.២ ប្រវត្តិរបស់ Microsoft Visual Studio(VS Code)	13
១.៤.៣ ទ្រឹស្តី IntelliJ.....	13
១.៤.៤ ទ្រឹស្តី Git hub.....	14

ជំពូកទី២ ការវិភាគគម្រោង

២.១ វិធីសាស្ត្រនៃការស្រាវជ្រាវ	15
២.១.១ ជំហាននៃការស្រាវជ្រាវ.....	15
២.២ រចនាសម្ព័ន្ធនៃការស្រាវជ្រាវ.....	16
២.២.១ ការរៀបចំគម្រោង(Planning & Requirements).....	16
២.២.២ ការវិភាគ (Analysis)	17
២.២.៣ ការរចនា (Design)	18
២.២.៤ ការអភិវឌ្ឍន៍ (Implementation)	18
២.២.៥ ការធ្វើតេស្តសាកល្បង (Testing).....	18
២.២.៦ ការថែទាំ (Maintenance).....	19
២.៣ វិធីសាស្ត្រប្រមូលទិន្នន័យ	19
២.៣.១ ទិន្នន័យចម្បង (Primary Data)	19
២.៣.២ ទិន្នន័យបន្ទាប់បន្សំ (Secondary Data).....	19
២.៤ គម្រោងពេលវេលានៃការស្រាវជ្រាវ.....	19

ជំពូកទី៣ ការវិភាគទិន្នន័យ ការគ្រប់គ្រង និងការអនុវត្ត

៣.១ គម្រោងរចនាសម្ព័ន្ធនៃការតម្លើងប្រព័ន្ធ	21
៣.១.១ គម្រោងទិដ្ឋភាពរចនាសម្ព័ន្ធ	21
៣.២ ការតម្លើងប្រព័ន្ធគម្រោង.....	21
៣.២.១ តម្រូវការ (Requirement).....	21
៣.២.២ ការវិភាគប្រព័ន្ធ (Analysis)	24
៣.២.៣ ការរចនាប្រព័ន្ធ(Design)	25
៣.២.៤ ការអភិវឌ្ឍន៍ (Implementation)	32

សេចក្តីសន្និដ្ឋាន និងអនុសាសន៍

១. សេចក្តីសន្និដ្ឋាន	38
២. អនុសាសន៍.....	39

ឯកសារយោង

ឧបសម្ព័ន្ធ

បញ្ជីតារាង

តារាងទី១ ៖ គម្រោងនៃការស្រាវជ្រាវ	20
តារាងទី១ ៖ Data Dictionary	29

បញ្ជីដ្យាក្រាម

ដ្យាក្រាមទី១ ៖ សម្រាប់ Teacher	22
ដ្យាក្រាមទី២ ៖ សម្រាប់ Student.....	22
ដ្យាក្រាមទី៣ ៖ System Requirement.....	23
ដ្យាក្រាមទី៤ ៖ Data relationship.....	26
ដ្យាក្រាមទី៥ ៖ ERD Diagram.....	30
ដ្យាក្រាមទី៦ ៖ Software Architecture	31

បញ្ជីរូបភាព

រូបភាពទី១ ៖ Logo Google Classroom	2
រូបភាពទី២ ៖ Logo Zoom Meeting	2
រូបភាពទី៣ ៖ Logo Google Meet.....	2
រូបភាពទី៤ ៖ Logo Tailwind CSS	8
រូបភាពទី៥ ៖ Logo Next Js	8
រូបភាពទី៦ ៖ Logo Javascrip	9
រូបភាពទី៧ ៖ Logo Spring Boot	9
រូបភាពទី៨ ៖ System Development Life Cycle (SDLC).....	11
រូបភាពទី៩ ៖ Logo PostgreSQL	12
រូបភាពទី១០ ៖ Logo Microsoft.....	12
រូបភាពទី១១ ៖ Logo Visual Studio Code	13
រូបភាពទី១២ ៖ Logo IntelliJ.....	14
រូបភាពទី១៣ ៖ Git Hab	14
រូបភាពទី១៤ ៖ ជំហាននៃការស្រាវជ្រាវ	15
រូបភាពទី១៥ ៖ Tree Structure.....	21
រូបភាពទី១៦ ៖ Login Form.....	32

រូបភាពទី១៧ ៖ Registration Form	33
រូបភាពទី១៨ ៖ New Homepage	33
រូបភាពទី១៩ ៖ After Join Class	34
រូបភាពទី២០ ៖ Announcement Dashboard.....	34
រូបភាពទី២១ ៖ Classwork Dashboard.....	35
រូបភាពទី២២ ៖ Materials Dashboard	35
រូបភាពទី២៣ ៖ Member Dashboard.....	36
រូបភាពទី២៤ ៖ User Setting-Setting	36
រូបភាពទី២៥ ៖ Class Setting-Setting.....	37

សេចក្តីផ្តើម

១. លំនាំបញ្ជាក់នៃការស្រាវជ្រាវ

ឆ្នាំ២០២៤ ពិភពលោកកាន់តែមានភាពរីកចម្រើនទៅមុខឥតឈប់ឈរដោយបច្ចេកវិទ្យាមានការអភិវឌ្ឍន៍ ទំនើប ពីមួយជំហានទៅមួយជំហានធ្វើឱ្យមនុស្សមានភាពងាយស្រួលក្នុងការរស់នៅប្រកបដោយជោគជ័យសុខភាពនិរន្តរភាព ។ល។ បច្ចេកវិទ្យាបានជួយសម្រួលស្ថានភាពសេដ្ឋកិច្ច ជាពិសេសនៅកំឡុងពេលដែលពិភពលោក កំពុងតែ ជួបបញ្ហាជាច្រើនមាន ជំងឺឆ្លង សង្គ្រាម គ្រោះរាំងស្ងួត បញ្ហាការប្រែប្រួលនៃអាកាសធាតុ។ ជាក់ ស្តែងនៅចុងឆ្នាំ ២០១៩ ដល់ ២០២១ ពិភពលោកបានជួបនូវជំងឺរីករាលដាល មួយប្រភេទដែលអ្នក វិទ្យាសាស្ត្របានឱ្យឈ្មោះថា កូវីដ-១៩(Covid-19) ក្នុងនោះគេបានឃើញថាបច្ចេកវិទ្យាបានដើរតួយ៉ាងសំខាន់ ក្នុងការដោះស្រាយនៅបញ្ហាប្រឈមមួយគឺ វិស័យអប់រំ ដែលអាចឱ្យសិស្សានុសិស្សបានបន្តការសិក្សា ប្រកបដោយភាពជោគជ័យ ទោះបីជាមានការបិទមិនឱ្យមនុស្សមានការជួបជុំក៏ដោយ លើសពីនេះទៅទៀត ស្ថានភាពសេដ្ឋកិច្ចក៏ត្រូវបានបច្ចេកវិទ្យាជួយសម្រួលឱ្យមានស្ថេរភាពនៅមួយកម្រិតនៅតាមបណ្តាប្រទេសកំពុងអភិវឌ្ឍ និងខ្លាំងក្លានៅបណ្តាប្រទេសអភិវឌ្ឍន៍។ ក្រសួងអប់រំចាប់ផ្តើមធ្វើការអោយមានកម្មវិធីសិក្សានៅលើប្រព័ន្ធអ៊ីនធឺណិត (Internet)ដោយសិស្សានុសិស្ស និស្សិតពុំចាំបាច់ទៅសាលា ដោយគ្រាន់តែសិក្សានៅលើ platform ដែលជាកម្មវិធីសិក្សាតាមបច្ចេកវិទ្យាជាការស្រេច ។

ដោយហេតុនេះហើយ ទើបក្រុមយើងខ្ញុំបានសម្រេចចិត្តធ្វើការស្រាវជ្រាវលើប្រធានបទមួយ គឺ **ការគ្រប់គ្រងប្រព័ន្ធថ្នាក់រៀនតាមបែបឌីជីថល** ដើម្បីជំនួយសម្រាប់ភាពងាយស្រួលដល់សិស្សានុសិស្សនិង អ្នកសិក្សាស្រាវជ្រាវអាចបន្ថែមចំណេះដឹង លើការសិក្សាតាមប្រព័ន្ធអេឡិចត្រូនិច។

២. ចំណេញបញ្ជាក់

ដូចដែលបានឃើញស្រាប់ហើយថា នៅក្នុងកំឡុងពេលមានបញ្ហាកូវីដ១៩ (COVID-19) សាលារៀនគ្រប់កម្រិតទាំងអស់នៅទូទាំងប្រទេសត្រូវបានបិទទ្វារ ហើយសិស្សានុសិស្សទាំងអស់ត្រូវបែរមកសិក្សាតាមរយៈប្រព័ន្ធអេឡិចត្រូនិច ដោយប្រើប្រាស់កម្មវិធីដូចជា Google Classroom , Zoom Meeting និង Google Meeting ជាដើមជំនួសវិញ។ ក្រៅពីនេះក៏មាននូវសិស្សាបុសិស្សមួយចំនួនធំ បានងាកមកប្រើប្រាស់នូវគេហទំព័រ រៀនពីចម្ងាយ (E-Learning) ហើយចំណេះក្រសួងអប់រំយុវជន និងកីឡា ក៏បានងាកមកប្រើប្រាស់ ឬសហការជាមួយគេហទំព័រណា ដែលមានមុខងាររៀនពីចម្ងាយ ដើម្បីជួយសម្រួល ក៏ដូចជាជួយបន្ថែមដល់ការសិក្សារបស់សិស្សានុសិស្ស អោយកាន់តែមានប្រសិទ្ធភាពជាងមុន នៅខណៈពេលមានកូវីដ១៩។



រូបភាពទី១៖ Logo របស់ Google Classroom



រូបភាពទី២៖ Logo របស់ Zoom Meeting



រូបភាពទី៣៖ Logo របស់ Google Meet

៣.គោលបំណង

គោលបំណងនៅក្នុងការស្រាវជ្រាវនេះ ក្រុមរបស់យើងខ្ញុំមានបំណងចង់លើកស្ទួយនៅវិស័យមួយនេះ និងផ្តល់ភាពទាន់សម័យនិយមទៅកាន់សិស្សានុសិស្សនិង ប្រជាជនកម្ពុជា ជា ពិសេសទៅលើការអភិវឌ្ឍន៍វិស័យអប់រំ ។

ក្រុមយើងខ្ញុំបានបង្កើតនៅ គេហទំព័រមួយ ដែលអាចជួយសម្រួលដល់ការសិក្សារបស់សិស្សានុសិស្ស ដោយមានភាពងាយស្រួល ដែលអាចសិក្សាបាននៅគ្រប់ទីកន្លែងនិងគ្រប់ពេលវេលាដែលពួកគេចង់សិក្សា។ ហើយកម្មវិធីនេះក៏បានផ្តល់ភាពងាយស្រួល ទៅកាន់ លោកគ្រូ អ្នកគ្រូ និងសិស្សានុសិស្សគ្រប់មជ្ឈដ្ឋានផងដែរ។

៤.វិសាលភាពនិងដែនកំណត់សិក្សា

ការស្រាវជ្រាវនេះផ្ដោតសំខាន់ទៅលើការសិក្សា និងគ្រប់គ្រងថ្នាក់រៀនតាមបែបឌីជីថលដែលប្រើប្រាស់ តាមរយៈប្រព័ន្ធអ៊ីនធឺណិតដែលមានវិសាលភាពនិងដែនកំណត់ដូចខាងក្រោម៖

- ការបង្កើត កិច្ចការ កម្រងសំណួរ ការប្រលង
- ការដាក់ពិន្ទុលើកិច្ចការ

- ការផ្តល់យោបល់លើកិច្ចការរបស់សិស្ស
- សិស្សអាច មើលកិច្ចការក្នុងថ្នាក់ ហើយបញ្ជូនវាទៅគ្រូរបស់ពួកគេ
- មើលពិន្ទុ និងមតិកែលម្អពីគ្រូ

៥.សារៈសំខាន់នៃការសិក្សា

ប្រធានបទនេះពិតជាមានសារៈសំខាន់ណាស់ សម្រាប់អ្នកដែលចង់សិក្សាទៅលើប្រព័ន្ធសិក្សាបែបឌីជីថល។ជាពិសេសតាមរយៈប្រធានបទមួយនេះនិងបង្ហាញពីមុខងារជាច្រើនដែលមាននៅក្នុងប្រព័ន្ធ និងធ្វើអោយអ្នកសិក្សាបានយល់ដឹងនិងស៊ីជម្រៅ។

៦.វិធីសាស្ត្រនៃការស្រាវជ្រាវ

នៅមុនពេលដែលពួកយើងបានធ្វើការពិភាក្សាគ្នាទៅលើប្រធានបទមួយនេះ ពួកយើងបានគិតទៅដល់ នៅបញ្ហាធំមួយដែលបញ្ហានោះគឺជាការប្រមូលទិន្នន័យផ្សេងៗដែលពាក់ព័ន្ធទៅនឹងប្រធានបទ។ ដោយ ឡែកទិន្នន័យត្រូវបានបែងចែកជាពីរ គឺ primary data និង secondary data។

- ❖ ទិន្នន័យចម្បង (primary data) ជាទិន្នន័យដែលបានមកពីរ
 - ការសាកសួរទៅកាន់លោកគ្រូអ្នកគ្រូនៅក្នុងសាលា
 - ការសាកសួរទៅកាន់រៀបចំផ្សេង
- ❖ ទិន្នន័យបន្ទាប់បន្សំ(secondary data) ជាទិន្នន័យដែលបានមកពី
 - ការស្វែងរកនៅលើអ៊ីនធឺណែត
 - ឯកសាររបស់លោកគ្រូអ្នកគ្រូ ឬសិស្សច្បងដែលបានបន្សល់ទុកនៅបណ្ណាល័យ

៧.បទសម្ព័ន្ធនៃការស្រាវជ្រាវ

ការសិក្សាស្រាវជ្រាវនេះចែកជា ៣ ជំពូកគឺ៖

ជំពូកទី១ ៖ រំលឹកទ្រឹស្តីដែលពាក់ព័ន្ធ (រៀបរាប់អំពីបរិបទទូទៅដូចជាប្រវត្តិ និងទ្រឹស្តីផ្សេងៗដែលពាក់ព័ន្ធ ក្នុងការបង្កើត Website។

ជំពូកទី២ ៖ ការវិភាគគម្រោង (រៀបរាប់អំពីវិធីសាស្ត្រនៃការស្រាវជ្រាវ និងពេលវេលានៃការនៃការស្រាវជ្រាវគម្រោង។

ជំពូកទី៣ ៖ ការវិភាគទិន្នន័យ (រៀបរាប់អំពីការប្រើប្រាស់ នឹងដំណើរការនៃ គម្រោង។

ជំពូកទី១

រំលឹកទ្រឹស្តីពាក់ព័ន្ធ

១.១.អំពីគេហទំព័រ

១.១.១.តើអ្វីទៅជាគេហទំព័រ ?

គេហទំព័រគឺជាបណ្តុំនៃ Web page ដែលចូលទៅប្រើបានតាមរយៈអ៊ីនធឺណិត ហើយជាធម្មតាវាទាក់ទងទៅនឹងប្រធានបទ ឬគោលបំណងជាក់លាក់ណាមួយ។ វាអាចមានអត្ថបទ រូបភាព វីដេអូ និងធាតុ ពហុព័ត៌មានផ្សេងទៀត ហើយអាចចូលប្រើបានដោយប្រើកម្មវិធីរកកម្រិតតាមអ៊ីនធឺណិត។ គេហទំព័រត្រូវបានប្រើ ប្រាស់សម្រាប់គោលបំណងផ្សេងៗដូចជា ការប្រើប្រាស់ក្នុងប្រព័ន្ធអប់រំ ការផ្សព្វផ្សាយព័ត៌មាន ពាណិជ្ជកម្មអេឡិចត្រូនិក បណ្តាញសង្គម និងការកម្សាន្តផ្សេងៗ។

គេហទំព័រគឺជាឯកសារឌីជីថលដែលត្រូវបានសរសេរដោយប្រើ HTML, CSS and JavaScript ដែលជាភាសាគោល។ រួចបង្ហាញនៅលើកុំព្យូទ័រដែលភ្ជាប់នឹងអ៊ីនធឺណិត កុំព្យូទ័របែបនេះត្រូវបានគេស្គាល់ថាជា Web Server គេហទំព័រជាសំណុំនៃ Webpage ពាក់ព័ន្ធដែលមានទីតាំងនៅក្រោម Domain name មួយ ដែលជាធម្មតាត្រូវបានបង្កើតដោយមនុស្សតែម្នាក់ ឬអង្គភាពមួយ។

គេហទំព័រគឺជាបណ្តុំនៃ Web page ដែលស្ថិតនៅក្រោម Domain name តែមួយហើយអ្នកអភិវឌ្ឍន៍បើកចេញវិទ្យាផ្សេងៗ ដើម្បីឱ្យដំណើរការឡើងដោយផ្តល់ភាពងាយស្រួលទៅដល់ User ក្នុងការស្វែងរកនៅប្រធានបទដែលពួកគេចង់បាន។

១.១.២. ប្រភេទនៃគេហទំព័រ

គេហទំព័រមានពីរប្រភេទគឺ Static Website និង Dynamic Website ហើយវាមាន គោលបំណង ផ្សេងគ្នា និងត្រូវបានបង្កើតឡើងដោយប្រើបច្ចេកវិទ្យាផ្សេងៗគ្នា។

Static Website គឺជាប្រភេទគេហទំព័រដែលមានមាតិកាថេរ ហើយមាតិការមានការជួសជុលលុះត្រាតែធ្វើបច្ចុប្បន្នភាព ដោយអ្នកអភិវឌ្ឍន៍គេហទំព័រ។ គេហទំព័រទាំងនេះត្រូវបានបង្កើតដោយប្រើ HTML, CSS និងពេលខ្លះ JavaScript។ ខ្លឹមសារនៅលើគេហទំព័រ Static នេះត្រូវបានកំណត់ជាមុន និងបង្ហាញដល់អ្នកទស្សនាទាំងអស់តាមរបៀបដូចគ្នា។

❖ គុណសម្បត្តិនៃ Static Website ៖

- ងាយស្រួលក្នុងការអភិវឌ្ឍ និងបង្ហាញ ទាមទារធនធានម៉ាស៊ីនមេតិចបំផុត។

- ពេលវេលាផ្ទុកលឿនព្រោះមិនចាំបាច់ទាញយកទិន្នន័យពីមូលដ្ឋានទិន្នន័យ។
- កាន់តែមានសុវត្ថិភាព ដោយសារមានអន្តរកម្មមានកម្រិតជាមួយបច្ចេកវិទ្យាខាងម៉ាស៊ីនមេ។
- ❖ គុណវិបត្តិនៃគេហទំព័រ Dynamic ៖
 - អន្តរកម្មមានកម្រិត និងខ្លីមសារStatic។
 - ការធ្វើបច្ចុប្បន្នភាពមាតិកាទាមទារចំណេះដឹងបច្ចេកទេស ឬជំនួយពីអ្នកអភិវឌ្ឍន៍។
 - មិនស័ក្តិសមសម្រាប់គេហទំព័រដែលមានការផ្លាស់ប្តូរញឹកញាប់ ឬមាតិកាជាក់លាក់របស់អ្នកប្រើប្រាស់។

Dynamic Website ប្រភេទនៃគេហទំព័រដែលប្រើស្ត្រីបង្កើតខាងម៉ាស៊ីនមេ និងមូលដ្ឋានទិន្នន័យដើម្បីបង្កើតមាតិកាភ្លាមៗ។ នេះមានន័យថាខ្លឹមសារនៅលើគេហទំព័រDynamic អាចផ្លាស់ប្តូរអាស្រ័យលើអ្នកប្រើប្រាស់។ គេហទំព័រDynamicមានអន្តរកម្មច្រើនជាងមុន ហើយអាចរួមបញ្ចូលលក្ខណៈពិសេសដូចជាការចូលរបស់អ្នកប្រើប្រាស់ មុខងារពាណិជ្ជកម្មអេឡិចត្រូនិក និងមាតិកាផ្ទាល់ខ្លួន។ ពួកវាជាធម្មតាមានភាពស្មុគស្មាញជាងគេហទំព័រ Static ហើយត្រូវការអ្នកជំនាញជំនាញសរសេរកម្មវិធីកម្រិតខ្ពស់បន្ថែមទៀតដើម្បីបង្កើត និងថែទាំ។ គេហទំព័រDynamic ត្រូវបានប្រើប្រាស់ជាញឹកញាប់សម្រាប់អាជីវកម្មធំៗ ហាងអនឡាញ បណ្តាញទំនាក់ទំនងសង្គម និងគេហទំព័រផ្សេងទៀតដែលតម្រូវឱ្យមានការអាប់ដេតញឹកញាប់ ឬមុខងារស្មុគស្មាញ។

- ❖ អត្ថប្រយោជន៍នៃគេហទំព័រ Static ៖
 - អន្តរកម្មខ្ពស់ជាមួយនឹងលក្ខណៈពិសេសដែលងាយស្រួលប្រើដូចជាប្រព័ន្ធចូល មាតិកាផ្ទាល់ខ្លួន និងមុខងារស្វែងរកStatic។
 - ការគ្រប់គ្រងមាតិកាកាន់តែងាយស្រួលតាមរយៈឧបករណ៍ CMS (ប្រព័ន្ធគ្រប់គ្រងមាតិកា)។
 - ស័ក្តិសមសម្រាប់គេហទំព័រខ្នាតធំដែលមានទិន្នន័យផ្លាស់ប្តូរញឹកញាប់។
- ❖ គុណវិបត្តិនៃគេហទំព័រStatic៖
 - កាន់តែស្មុគស្មាញក្នុងការអភិវឌ្ឍន៍ និងថែទាំបើប្រៀបធៀបទៅនឹងគេហទំព័រDynamic។
 - តម្រូវការធនធានម៉ាស៊ីនមេ និងមូលដ្ឋានទិន្នន័យកើនឡើង ដែលនាំឱ្យតម្លៃបង្ហោះមានសក្តានុពលខ្ពស់។
 - ពេលវេលាផ្ទុកយឺតជាងបើប្រៀបធៀបទៅនឹងគេហទំព័រDynamic ដោយសារដំណើរការផ្ទុកខាងម៉ាស៊ីនមេ។

១.១.៣ .តើអ្វីទៅជា Digital Classroom ?

Digital Classroom គឺជាបរិយាកាសសិក្សាមួយ ដែលមាតិកាអប់រំត្រូវបានផ្តល់ជូនតាមរយៈវេទិកាឌីជីថល និងឧបករណ៍ផ្សេងៗ ជំនួសវិញនឹងការបង្រៀនផ្ទាល់មុខជាប្រពៃណី។ វាជាញឹកញាប់ប្រើកុំព្យូទ័រ ឬទូរស័ព្ទដៃ ដើម្បីចូលប្រើមេរៀនតាមអ៊ីនធឺណិត ការងារផ្តល់ និងធនធានសិក្សាផ្សេងៗទៀត។

Digital Classroom អាចជាការសិក្សាតាមអ៊ីនធឺណិតទាំងស្រុង ឬជាសិស្សានុសិស្សដែលប្លង់ដោយប្រើប្រាស់ប្រព័ន្ធអ៊ីនធឺណិតជាមួយនឹងការបង្រៀនផ្ទាល់មុខ។

១.១.៤. ប្រភេទនៃគេហទំព័រ Digital Classroom

មានប្រភេទគេហទំព័រនិងវេទិកាជាច្រើនដែលអាចប្រើជាកម្មវិធី Digital Classroom:

- ប្រព័ន្ធគ្រប់គ្រងការសិក្សា (LMS): វេទិកាដូចជា Moodle, Canvas, និង Google Classroom ដែលផ្តល់ឧបករណ៍ពេញលេញសម្រាប់គ្រប់គ្រងមុខវិជ្ជា រួមមានការធ្វើផែនការមេរៀន ការផ្តល់ការងារ និងការវាយតម្លៃ។
- ឧបករណ៍វីដេអូក្រុមប្រឹក្សាសាលា: ឧបករណ៍ដូចជា Zoom, Microsoft Teams, និង Google Meet ដែលអនុញ្ញាតឱ្យមានថ្នាក់អនុញ្ញាតផ្ទាល់។
- វេទិកាធនធានសិក្សា: គេហទំព័រដូចជា Khan Academy ឬ Coursera ដែលផ្តល់មាតិកាសិក្សានិងមុខវិជ្ជា។
- -ឧបករណ៍សហការជាក្រុម: វេទិកាដូចជា Padlet, Trello, ឬ Miro ដែលជួយអោយមានគម្រោងក្រុម និងការសិក្សាសហការជាក្រុម។

១.១.៥.ប្រវត្តិនៃការបង្កើត Digital Classroom

ដើមឆ្នាំ 2000: ការណែនាំប្រព័ន្ធគ្រប់គ្រងការសិក្សា (LMS) ដូចជា Blackboard និង Moodle គឺជាការចាប់ផ្តើមនៃការអប់រំតាមអ៊ីនធឺណិតដែលមានចន្លោះសម្ព័ន្ធ។

ឆ្នាំ 2010: ការកើនឡើងនៃ MOOCs (ការបង្រៀនមុខវិជ្ជាជាច្រើនតាមអ៊ីនធឺណិត) ដោយវេទិកាដូចជា Coursera និង edX បានពង្រីក Digital Classroom ទៅកម្រិតសកល។

ឆ្នាំ 2020: ការរាតត្បាតនៃជំងឺ COVID-19 បានកាន់តែរុះរើអោយមានការប្រើប្រាស់ Digital Classroom ជាសកល ដែលនាំឱ្យមានការប្រើប្រាស់ទូទៅនៃវេទិកាដូចជា Zoom និង Google Classroom សម្រាប់ការសិក្សាដំណាក់កាល K-12 និងការអប់រំកម្រិតខ្ពស់។

១.១.៦. គោលបំណងក្នុងការបង្កើត Digital Classroom

Digital Classroom ត្រូវបានបង្កើតឡើងដើម្បីបង្កើន និងផ្លាស់ប្តូរប្រព័ន្ធអប់រំបែបប្រពៃណី ប្រើប្រាស់បច្ចេកវិទ្យាដើម្បីកែលម្អលទ្ធភាពប្រើប្រាស់ ភាពបត់បែន និងបទពិសោធន៍សិក្សាទាំងមូល។ នេះគឺជាហេតុផលសំខាន់ៗមួយចំនួនដែល Digital Classroom ត្រូវបានបង្កើតឡើង៖

- ❖ លទ្ធភាពប្រើប្រាស់៖ ថ្នាក់រៀនឌីជីថលអនុញ្ញាតឱ្យសិស្សមកពីទីតាំងភូមិសាស្ត្រផ្សេងៗទទួលបានការអប់រំប្រកបដោយគុណភាព។ នេះមានប្រយោជន៍ជាពិសេសសម្រាប់សិស្សានុសិស្សនៅតំបន់ដាច់ស្រយាល ឬតំបន់ក្រីក្រ ដែលសាលារៀនមានខ្វះខាត។
- ❖ ភាពបត់បែន៖ ផ្តល់នូវភាពបត់បែនក្នុងការរៀនតាមល្បឿន និងពេលវេលាផ្ទាល់ខ្លួន។ វាមានប្រយោជន៍ជាពិសេសសម្រាប់អ្នកសិក្សាពេញវ័យ អ្នកជំនាញធ្វើការ ឬសិស្សដែលមានការប្តូរធាតុផ្សេងទៀត។
- ❖ ការរៀនតាមបុគ្គល៖ ថ្នាក់រៀនឌីជីថលអាចប្រើប្រាស់ទិន្នន័យ និងបច្ចេកវិទ្យាសម្របខ្លួន ដើម្បីសម្រួលបទពិសោធន៍សិក្សាទៅតាមតម្រូវការរបស់សិស្សម្នាក់ៗ ដោយជួយឱ្យអ្នកសិក្សាម្នាក់ៗរីកចម្រើនក្នុងល្បឿនរបស់ពួកគេផ្ទាល់។
- ❖ លទ្ធភាពធ្វើមាត្រដ្ឋាន៖ ថ្នាក់រៀនឌីជីថលអាចផ្ទុកសិស្សបានយ៉ាងច្រើនក្នុងពេលតែមួយ ដែលធ្វើឱ្យអាចទៅដល់ និងអប់រំមនុស្សកាន់តែច្រើននៅទូទាំងពិភពលោក។
- ❖ ប្រសិទ្ធភាពនៃការចំណាយ៖ តាមរយៈការកាត់បន្ថយតម្រូវការហេដ្ឋារចនាសម្ព័ន្ធរូបវន្ត ការធ្វើដំណើរ និងសម្ភារៈបោះពុម្ព ថ្នាក់រៀនឌីជីថលអាចកាត់បន្ថយថ្លៃដើមនៃការអប់រំសម្រាប់ទាំងស្ថាប័ន និងសិស្ស។
- ❖ ភាពធន់៖ ថ្នាក់រៀនឌីជីថលផ្តល់នូវប្រព័ន្ធអប់រំដែលធន់ជាងមុន ជាពិសេសក្នុងស្ថានភាពដូចជាជំងឺរាតត្បាត ឬគ្រោះមហន្តរាយធម្មជាតិ ដែលការកំណត់ថ្នាក់រៀនបែបប្រពៃណីអាចនឹងមានការរំខាន។

១.២. ទ្រឹស្តីពាក់ព័ន្ធការបង្កើត DCMS

១.២.១ Tailwind CSS

Tailwind CSS គឺជាកម្មវិធីជំនួយដំបូង CSS ក្របខ័ណ្ឌដែលបានរចនាឡើងដើម្បីពន្លឿនដំណើរការនៃការកំណត់រចនាប័ទ្មគេហទំព័រ។ ជំនួសឱ្យការសរសេរផ្ទាល់ខ្លួន CSS អ្នកអភិវឌ្ឍន៍ប្រើថ្នាក់ឧបករណ៍ប្រើប្រាស់ដែលបានកំណត់ជាមុនដោយផ្ទាល់នៅក្នុង HTML របស់ពួកគេដើម្បីអនុវត្តរចនាប័ទ្ម។ វិធីសាស្ត្រនេះលើកកម្ពស់ការរចនាដែលជាប់លាប់ជាងមុន និងបង្កើនល្បឿននៃការអភិវឌ្ឍន៍ដោយកាត់បន្ថយតម្រូវការក្នុងការប្តូររវាងឯកសារ CSS និង HTML។



Tailwind CSS

រូបភាពទី៤៖ Logo របស់ Tailwind CSS

១.២.២ Next JS

Next.js គឺជា React Framework ដែលអាចឱ្យអ្នកអភិវឌ្ឍន៍បង្កើតកម្មវិធីគេហទំព័របានលឿន និង អាចធ្វើមាត្រដ្ឋានបាន ជាមួយនឹងលក្ខណៈពិសេសដូចជាការបង្ហាញផ្នែកខាងម៉ាស៊ីនមេ ការបង្កើតគេហទំព័រភីអិល វេន វេន API និងការបំបែកកូដដោយស្វ័យប្រវត្តិ។ វាជួយសម្រួលដល់ការអភិវឌ្ឍន៍គេហទំព័រទាំងភីអិល វេន និងថាមវន្ត ដោយផ្តល់នូវរចនាសម្ព័ន្ធដ៏រឹងមាំ ដំណើរការប្រសើរឡើង និងសមត្ថភាព SEO ដែលប្រសើរឡើង។



រូបភាពទី៥៖ Logo របស់ Next.js

១.២.៣ JavaScript

JavaScript គឺជាclient site script មួយសម្រាប់គេហទំព័រ គឺវាមិនត្រូវការ Server ដើម្បីធ្វើការ Compile កូដរបស់វានោះទេ វាអាចដំណើរការលើ Browser តែម្តង។

JavaScript គឺជាភាសាដ៏មានប្រជាប្រិយភាពមួយនៅលើប្រព័ន្ធអ៊ីនធឺណិត និង វាដំណើរការនៅលើ browsers ជាច្រើនដូចជា Internet explorer, Firefox, Chrome, Opera, Safari និង browsers ជាច្រើន ផ្សេងទៀត។

JavaScript



រូបភាពទី៦៖ Logo របស់ JavaScript

១.២.៤ SpringBoot

Spring Boot ជាការ Bootstrapping នូវ Project របស់ **Spring** ហើយ **Spring** គឺជា framework របស់ភាសា **Java** ដែលប្រើសម្រាប់សរសេរកូដបង្កើតជា Web Application ។



រូបភាពទី៧៖ Logo របស់ SpringBoot

១.៣ ទ្រឹស្តីពាក់ព័ន្ធនឹង Database

១.៣.១ និយមន័យ Database

Database គឺជាការប្រមូលផ្តុំ និងគ្រប់គ្រងទិន្នន័យទាំងអស់ដែលទាក់ទងទៅនឹង ប្រធានបទមួយ ឬគំរោងណាមួយដូចជាការប្រមូលផ្តុំនិងគ្រប់គ្រងការខ្ចីសៀវភៅនៅក្នុង បណ្ណាល័យការប្រមូលផ្តុំនិងគ្រប់គ្រងលើបុគ្គលិកនៃក្រុមហ៊ុន អង្គការ ឬក្រសួងណាមួយ ជាដើម ។

១.៣.២ អ្វីទៅជា Database Management System (DBMS) ?

ប្រព័ន្ធគ្រប់គ្រងមូលដ្ឋានទិន្នន័យ (DBMS) សំដៅលើបច្ចេកវិទ្យាសម្រាប់បង្កើតនិងគ្រប់គ្រង Database Management System។ DBMS គឺជាឧបករណ៍ Software ដើម្បីរៀបចំ (បង្កើតទាញយកធ្វើការ

កែប្រែ និងគ្រប់គ្រង) ទិន្នន័យនៅ DBMS ។ គោលបំណងចម្បងរបស់ DBMS គឺដើម្បីផ្គត់ផ្គង់មធ្យោបាយរក្សាទុកនិងទាញយកព័ត៌មាន DBMS ដែលងាយស្រួលនិងមានប្រសិទ្ធភាព។

១.៣.៣ ទ្រឹស្តី System Development Life Cycle (SDLC)

តាមរយៈសៀវភៅ System Analysis and Design របស់អ្នកនិពន្ធ KENNETH E. KENDALL និង JULIE E.KENDALL ដែលបានបោះពុម្ពនៅក្នុងឆ្នាំ២០១១ បានសរសេរថាដំណើរ ការរៀបចំនូវ Information System Development Project ត្រូវបានបង្កើតឡើងតាមវិធីសាស្ត្រមួយ ដែលគេហៅថា System Development Life Cycle (SDLC)។ SDLC បានបង្ហាញនូវ គ្រប់ជំហាន ទាំងអស់នៃដំណាក់កាលការងាររបស់អ្នកឯកទេសខាងអភិវឌ្ឍន៍ប្រព័ន្ធ រួមមានអ្នកវិភាគប្រព័ន្ធ អ្នកចនាប្រព័ន្ធទិន្នន័យ និងអ្នកសរសេរកូដត្រូវអនុវត្តតាមដើម្បីអភិវឌ្ឍន៍ដាក់ឱ្យប្រើប្រាស់ ថែរក្សា និងអាច ជួសជុលនូវប្រព័ន្ធ។

វដ្តនៃការអភិវឌ្ឍន៍ប្រព័ន្ធ (SDLC) គឺជាដំណើរការដែលមានចន្លោះសម្រាប់ការធ្វើគម្រោង បង្កើតការធ្វើតេស្ត និងការដាក់ឱ្យដំណើរការប្រព័ន្ធព័ត៌មានឬកម្មវិធីផ្នែកទន់។ វាគឺជាចំណុចដំណាក់កាលដែលត្រូវអនុវត្តដើម្បីធានាថាគម្រោងត្រូវបានបញ្ចប់ដោយលំដាប់លំដោយ និងឆ្លើយតបនឹងតម្រូវការដែលបានកំណត់។ដំណើរការនេះត្រូវបានគេអះអាងថាជាជំហានដែលបន្ត និងបណ្តាក់គ្នា មានន័យថា ជំហានមួយចប់ ទើបអាចបន្តទៅដល់ជំហានមួយទៀត ហើយលទ្ធផលសម្រេចរបស់ជំហាននីមួយៗ គឺ ជាអ្វីដែលត្រូវបញ្ចូលទៅក្នុងជំហានបន្ទាប់ទៀត។ ប៉ុន្តែតាមសកម្មភាពជាក់ស្តែងដំណើរការនៃការអភិវឌ្ឍន៍ប្រព័ន្ធ (System Development Process) គឺមានទំនាក់ទំនងគ្នាទៅវិញទៅមក រួចហើយទើប បន្តទៅជំហានបន្ទាប់។

អ្នកវិភាគប្រព័ន្ធមិនអាចធ្វើការអភិវឌ្ឍន៍ដាច់ដាច់ថា តើក្នុងការបង្កើតប្រព័ន្ធដោយប្រើទ្រឹស្តី SDLC មានប៉ុន្មានជំហាន ឬប៉ុន្មានដំណាក់កាលនោះទេប្លែកគ្នាទៅតាមអ្នកនិពន្ធ ដែលខ្លះប្រើមានចំនួន ៥ ជំហាន/ដំណាក់កាល ខ្លះមាន ៦ជំហាន/ដំណាក់កាល និងមាន ៧ជំហាន/ដំណាក់កាល ហើយបើតាម ចរិតលក្ខណៈដែលពេញលេញរបស់ SDLC ត្រូវបានគេបែងចែងជា ៧ជំហាន/ដំណាក់កាល ។ ក៏ប៉ុន្តែតាមរយៈកាសិក្សានិងកំណត់យកជាគំរូតាងក្នុងទីនេះគឺយើងនឹងលើកយក SDLC ដែលបាន បែងចែងជា ៦ ដំណាក់កាល ។

វដ្តនេះគឺមានដំណាក់កាលខាងក្រោម៖

១. ការធ្វើគម្រោង (Planning & Requirements)

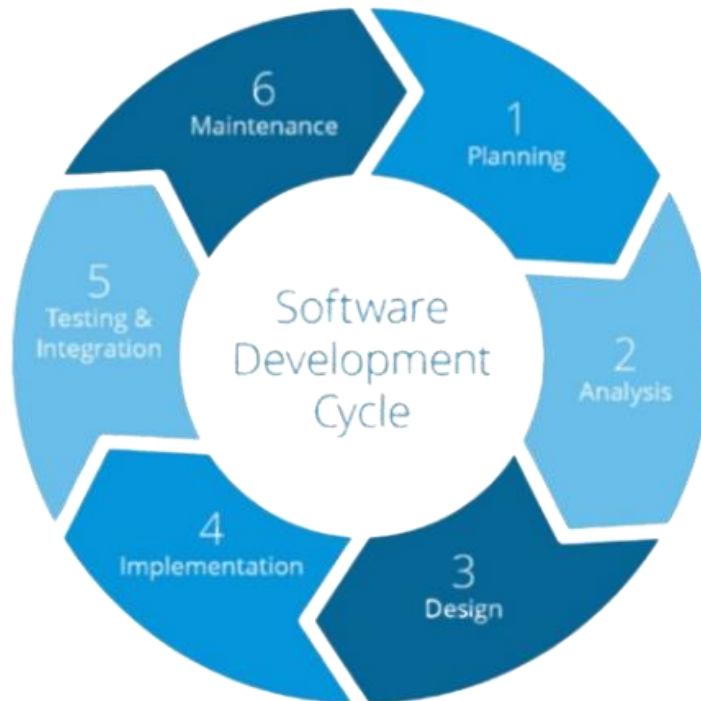
២.ការវិភាគ (Analysis)

៣.ការរចនា (Design)

៤.ការអភិវឌ្ឍន៍ (Implementation)

៥.ការធ្វើតេស្ត (Testing)

៦.ការថែទាំ (Maintenance)



រូបភាពទី៨៖ System Development Life Cycle (SDLC)

១.៣.៤ ទ្រឹស្តី PostgreSQL

PostgreSQL គឺជាប្រព័ន្ធគ្រប់គ្រងមូលដ្ឋានទិន្នន័យទំនាក់ទំនងវត្ថុ (ORDBMS) ដែលសរសេរជាភាសា C programming ដែលមានប្រភពបើកចំហ ឆ្លងវេទិកា និងឆ្លងវេទិកា។ វាត្រូវបានបង្កើតឡើងដំបូងនៅនាយកដ្ឋានវិទ្យាសាស្ត្រកុំព្យូទ័ររបស់សាកលវិទ្យាល័យកាលីហ្វ័រញ៉ា។ ទាំងភាសាសំណួរ SQL និង NoSQL ត្រូវបានគាំទ្រ។ វាអនុញ្ញាតឱ្យអ្នករក្សាទុកទិន្នន័យ និងសង្គ្រោះវាដោយសុវត្ថិភាពនៅពេលណាដែលសំណើត្រូវបានប្រតិបត្តិ។ វាអនុលោមតាម ACID គាំទ្រមុខងារ ដំណើរការរក្សាទុក និងកេះ ហើយមានភាពស្របគ្នាខ្លាំង។ វារួមបញ្ចូល MVCC (ការគ្រប់គ្រងស្របគ្នាច្រើនកំណែ) ដែលអនុញ្ញាតឱ្យអ្នកប្រើច្រើនធ្វើការលើប្រព័ន្ធក្នុងពេលតែមួយ។



រូបភាពទី៩៖ Logo របស់ PostgreSQL

១.៤ ឧបករណ៍និងវេបសាយជំនួយ

១.៤.១ ទ្រឹស្តីពាក់ព័ន្ធនឹង Microsoft

Microsoft គឺជាក្រុមហ៊ុនផលិតSoftware ធំបំផុតនៅលើសកលលោក។ផលិតផលដែលបង្កើតឡើង sum window operating system, Internet Explorer, Microsoft office, Microsoft Visual studio code និងកម្មវិធីបង្កើតApplicationផ្សេងៗជាច្រើនទៀត។ ក្នុងនោះ Visual studio code ត្រូវ បានបង្កើតឡើងដើម្បីឱ្យ developer អាចធ្វើការ develop Application បានដោយខ្លួនឯងនៃផ្នែកFront end របស់ Application។ នៅក្នុង Visual studio code អាចឱ្យ developerប្រើភាសាជាច្រើនដូចជា : C, C++, C#, Java, JavaScript, Node.js ,Next Js , Laravel , Futtter ជាដើម។



រូបភាពទី១០៖ Logo របស់ Microsoft

១.៤.២ ប្រភេទរបស់ Microsoft Visual Studio(VS Code)

Vs Code គឺជា Source code editor ដែលបង្កើតឡើងដោយ Microsoft ជាមួយនឹង Electron Framework ហើយអាចប្រើបានជាមួយ platform ជាច្រើនដូចជា Windows, Linux and macOS ។ Visual Studio Code ត្រូវបានប្រកាសអោយប្រើប្រាស់ដំបូងដោយ Microsoft នៅថ្ងៃទី 29 April 2015។ នៅ ថ្ងៃទី 18 November 2015 source របស់ Visual Studio code ត្រូវបានដាក់ក្រោមការគ្រប់គ្រងរបស់ MIT License នឹងអាចធ្វើការផ្ទុក Code នៅលើ GitHub បានផងដែរ។

Extension ដែល Support ក៏ត្រូវបាន ប្រកាសអោយប្រើប្រាស់នៅថ្ងៃទី 14 April 2015 ហើយបច្ចុប្បន្ន VS Code មាន Version 1.78។



រូបភាពទី១១៖ Logo របស់ Visual Studio Code

១.៤.៣ ទ្រឹស្តី IntelliJ

IntelliJ IDEA គឺជាការអភិវឌ្ឍន៍រួមបញ្ចូលគ្នា (IDE) ដែលបង្កើតឡើងដោយ JetBrains ។ វាត្រូវបានរចនាឡើងជាមួយសម្រាប់ការអភិវឌ្ឍន៍ Java ប៉ុន្តែក៏មានការគាំទ្រដ៏ទូលំទូលាយនៃភាសាសរសេរកម្មវិធីផ្សេងទៀត រួមមាន Kotlin, Groovy, Scala, Python និងច្រើនទៀត។ IntelliJ IDEA ផ្តល់នូវលក្ខណៈពិសេសដូចជាការបំពេញកូដឆ្លាតវៃ ការវិភាគកូដប្រព័ន្ធដ៏មានអានុភាព ឧបករណ៍ refactoring និងកម្មវិធីជំនួយផ្សេងៗ ដែលធ្វើឱ្យវាក្លាយជាជម្រើសដ៏ពេញនិយមក្នុងចំណោមអ្នកអភិវឌ្ឍន៍សម្រាប់ការសរសេរ សាកល្បង និងការកែកូដប្រកបដោយប្រសិទ្ធភាព។



រូបភាពទី១២៖ Logo របស់ IntelliJ

១.៤.៤ ទ្រឹស្តី Git hub

GitHub គឺជាសេវាកម្មដ៏ពេញនិយមបំផុតមួយសម្រាប់ការបង្ហោះកន្លែងផ្ទុកប្រភពបើកចំហ។ គេហទំព័រនេះអនុញ្ញាតឱ្យអ្នកបោះផ្សាយគម្រោងផ្ទាល់ខ្លួន និងតាមដានការផ្លាស់ប្តូរដែលបានធ្វើឡើងនៅក្នុងការផ្សាយឡើងវិញនីមួយៗ។ អ្នកប្រើប្រាស់ GitHub ផ្សេងទៀតអាចពិនិត្យមើលលេខកូដអ្នកប្រើប្រាស់ និងណែនាំការផ្លាស់ប្តូរផ្ទាល់ខ្លួនរបស់ពួកគេ។ និយមន័យនេះផ្តល់តែការយល់ដឹងខ្លីៗប៉ុណ្ណោះ។



រូបភាពទី១៣៖ Logo របស់ GitHab

ជំពូកទី២

ការវិភាគគម្រោង

២.១ វិធីសាស្ត្រនៃការស្រាវជ្រាវ

២.១.១ ជំហាននៃការស្រាវជ្រាវ

តាមរយៈការស្រាវជ្រាវរបស់ក្រុមយើងខ្ញុំដើម្បីលើកស្ទួយលើវិស័យអប់រំបែបឌីជីថលមួយនេះ ក្រុមយើងខ្ញុំបានជ្រើសរើសយកវិធីសាស្ត្រស្រាវជ្រាវ មួយឈ្មោះថា System Development Life Cycle (SDLC) មកធ្វើជាវិធីសាស្ត្រគំរូសម្រាប់ជាជំហាននីមួយៗនៃការវិភាគគម្រោងរបស់ក្រុមយើងខ្ញុំ។ ដូចដែលយើងខ្ញុំបានរៀបរាប់ និងរាយរាប់នៅនិយមន័យ ម្តងរួចមកហើយ។ ដូច្នេះហើយខាងក្រោមនេះ គឺជាជំហាននៃការស្រាវជ្រាវរបស់ក្រុមយើងខ្ញុំមានដូចតទៅ ៖

១.ការរៀបចំគម្រោង(Planning & Requirements)

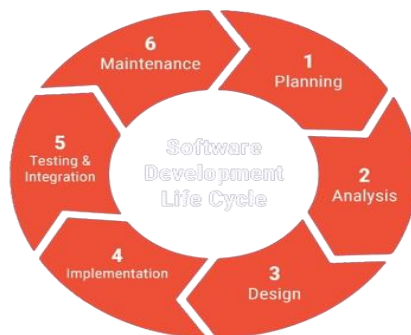
២.ការវិភាគ (Analysis)

៣.ការរចនា (Design)

៤.ការអភិវឌ្ឍន៍ (Implementation)

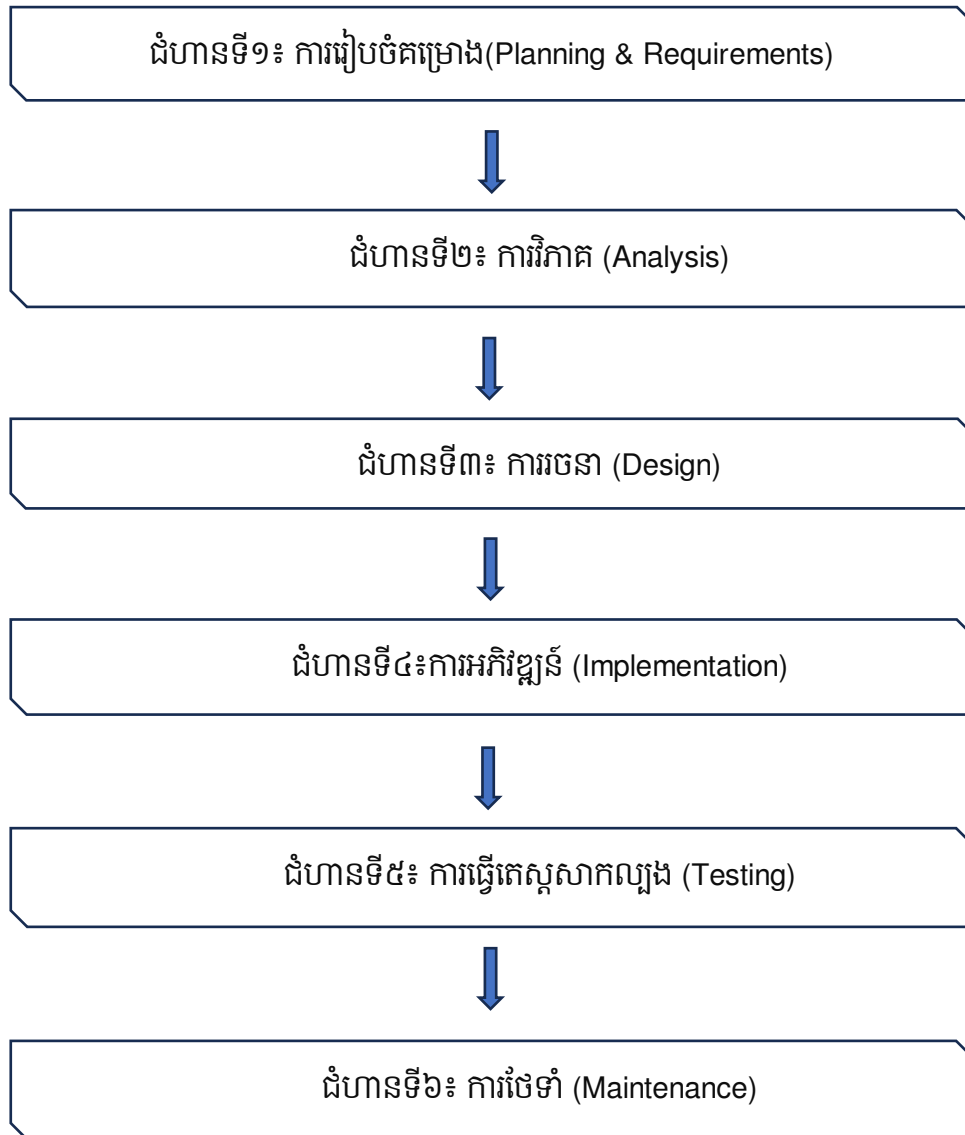
៥.ការធ្វើតេស្តសាកល្បង (Testing)

៦.ការថែទាំ (Maintenance)



រូបភាពទី១៤៖ ជំហាននៃការស្រាវជ្រាវ

២.២ បេសសម្ព័ន្ធនៃការស្រាវជ្រាវ



២.២.១ ការរៀបចំគម្រោង (Planning & Requirements)

ការរៀបចំគម្រោង (Planning & Requirements) គឺជាការកំណត់តម្រូវការសម្រាប់ប្រព័ន្ធថ្មីឬការផ្លាស់ប្តូរប្រព័ន្ធដែលមានស្រាប់។ ដំណាក់កាលនេះរួមមានការសិក្សាអំពីផលប៉ះពាល់ ការវិភាគនៃការចំណាយ និងការកំណត់ដែនកំណត់គម្រោង។

នៅក្នុងចំណុចនេះផងដែរ មុននឹងឈានទៅរកការជ្រើសរើសប្រធានបទ លើកយកមកសរសេរក្នុងនិក្ខេបបទនេះ យើងខ្ញុំបានពិភាក្សាគ្នា ដោយផ្តល់យោបល់គ្នាទៅវិញទៅមកយ៉ាងល្អិតល្អន់ ដើម្បីសម្រេចយក

ប្រធានបទមួយនេះ ដោយចូលរួមចំណែកសាស្ត្រាចារ្យដើម្បីដឹកនាំ និងផ្តល់យោបល់ល្អៗក្នុងការសរសេរ និក្ខេបបទនេះឡើង។

ក្រុមយើងខ្ញុំបានធ្វើការសិក្សាទៅលើលទ្ធភាពក្នុងការសាងសង់នូវគេហទំព័រនេះឡើង ដែលជួយ សម្រួលទៅដល់វិបាកដែល លោកគ្រូ អ្នកគ្រូ និង សិស្សានុសិស្សកំពុងជួបប្រទះបញ្ហា រួមមានដូចជា៖

- ✓ សិទ្ធិទទួលបានការអប់រំមិនស្មើគ្នា ជាពិសេសសិស្សជាច្រើននៅតំបន់ជាប់ស្រយាល
- ✓ សិស្សកង្វះការរៀនសូត្រផ្ទាល់ខ្លួន
- ✓ បន្ទប់រៀនកំណត់ចំនួនសិស្សដើម្បីចូលសិក្សា
- ✓ គ្រូបង្រៀនមានកំណត់ មិនអាចបង្រៀនសិស្សលើសចំនួនកំណត់ក្នុងថ្នាក់
- ✓ សិស្សមិនអាចទាយយកមេរៀនមកមើលមុនការបង្រៀនចូលមកដល់

២.២.២ ការវិភាគ (Analysis)

ការវិភាគ (Analysis) គឺជាការប្រមូល និងវិភាគតម្រូវការអាជីវកម្ម ឬកម្មវិធី និងបច្ចេកទេសសម្រាប់ ប្រព័ន្ធ។ វារួមមានការរក្សាទុកនៃឯកសារយ៉ាងលម្អិតអំពីអ្វីដែលប្រព័ន្ធត្រូវធ្វើ។

បន្ទាប់ពីពួកយើងខ្ញុំបានប្រមូលយកនូវទិន្នន័យ និងព័ត៌មានផ្សេងៗដែលពាក់ព័ន្ធ ដោយបានមកពី Internet រួចរាល់ ក្រុមយើងខ្ញុំបានធ្វើការវិភាគ និងកំណត់ទិន្នន័យដើម្បីបង្កើតប្រព័ន្ធនេះ ហើយស្វែងរកនូវ ចំណុចខ្វះខាតដើម្បីកំណត់បញ្ហា និងស្វែងរកដំណោះស្រាយឲ្យបានកាន់តែល្អប្រសើរដោយធ្វើការត្រួតពិនិត្យ ឡើងវិញ ដើម្បីធ្វើការវិភាគឲ្យបានត្រឹមត្រូវទៅនឹងគោលបំណង នៃការរៀបចំប្រព័ន្ធនេះ ឲ្យមានលក្ខណៈពេញ លេញសមស្របទៅនឹងតម្រូវការដែលយើងខ្ញុំបានកំណត់ទុក។

នៅក្នុងជំហាននៃការបង្កើតប្រព័ន្ធ នេះផងដែរ យើងខ្ញុំបានធ្វើការវិភាគប្រព័ន្ធដោយប្រើប្រាស់នូវ បច្ចេកទេសដូចជា៖

- Data Flow Diagram
- Database Normalization

នៅក្នុងជំហាននេះដែរ យើងបានជ្រើសយកនូវបច្ចេកទេសខាងលើ ដើម្បីជួយសម្រួល និងបង្កើនភាពងាយ ស្រួលដល់ការបង្កើតប្រព័ន្ធរបស់យើងខ្ញុំផងដែរ។

២.២.៣ ការរចនា (Design)

ដោយយើងខ្ញុំបានយល់ដឹង អំពីផលវិបាក និងតម្រូវការរបស់អ្នកប្រើប្រាស់ ជាហេតុនាំអោយយើងធ្វើការរចនាសម្ព័ន្ធនូវគេហទំព័រដែលយើងបានសិក្សាស្រាវជ្រាវធ្វើឡើង ប្រកបដោយភាពទាក់ទាញ សាមញ្ញ និងងាយស្រួលក្នុងការប្រើប្រាស់។ ពីព្រោះការរចនា គឺជាការរចនាសម្ព័ន្ធនៅក្នុងប្រព័ន្ធ រួមមានការរចនាប្រព័ន្ធទាំងមូល ក៏ដូចជាផ្នែកជាក់លាក់ដូចជាទិន្នន័យ កម្មវិធីប្រើប្រាស់ និងគំនូសណែនាំលំហូរទិន្នន័យ។

២.២.៤ ការអភិវឌ្ឍន៍ (Implementation)

ការអភិវឌ្ឍន៍ ឬការបង្កើត គឺជា ការសរសេរកូដ និងការបង្កើតប្រព័ន្ធតាមតម្រូវការ ការរចនា ដំណាក់កាលនេះរួមមានការសរសេរកម្មវិធី ការបញ្ចូលទិន្នន័យ និងការធ្វើតេស្តដំបូង។ ការសរសេរកូដ ជាដំណាក់កាលមួយដែលមានភាពស្មុគស្មាញ និងប្រើប្រាស់ពេលវេលាយូរជាងគេ ។ ដោយភាសាដែលត្រូវបានជ្រើសយកមកប្រើប្រាស់ក្នុងការសរសេរកូដនេះទៀតសោត បានជ្រើសយកនូវភាសា និង កម្មវិធីសម្រាប់សរសេរ ដូចទៅនឹងជំពូកទី១ ដែលជាជំពូករំលឹកទ្រឹស្តី។ គ្រប់ជំពូកដែលបានលើកយកមករំលឹក ទ្រឹស្តីសិទ្ធិសឹងតែជាភាសា និងកម្មវិធីដែលយើងខ្ញុំយកមកប្រើប្រាស់ក្នុងការសរសេរកូដ។ ក្នុងដំណាក់កាលសរសេរកូដនេះដែរ មិនមែនត្រឹមតែគ្រាន់តែចងបង្ហាញនូវលទ្ធផលប៉ុណ្ណោះទេ ប៉ុន្តែពួកយើងបានខិតខំប្រឹងប្រែងសរសេរកូដ បង្កើតនូវកម្មវិធីដែលបំពេញតម្រូវការ និងភាពខ្វះខាតនានាដែលអ្នកប្រើប្រាស់ គេហទំព័រតែងលើកឡើង។

២.២.៥ ការធ្វើតេស្តសាកល្បង (Testing)

ការតេស្តសាកល្បង គឺជាការត្រួតពិនិត្យទៅលើមុខងារ (Functions) ផ្សេងៗមុនពេលដែលយើង សម្រេចចិត្តដាក់ប្រព័ន្ធឲ្យអ្នកប្រើប្រាស់ដំណើរការជាផ្លូវការ ហើយរកនូវបញ្ហា Error ដែលមានភាពមិន ប្រក្រតីរបស់ប្រព័ន្ធ ឬ មុខងារផ្សេងទៀត។ ការ Test ប្រព័ន្ធនេះគឺបានធ្វើឡើងដោយក្រុមអ្នកសរសេរ កូដ និងក្រុមTesting ជាអ្នកប្រើប្រាស់ផ្ទាល់តែម្តង។ នេះគឺជាដំណាក់កាលដែល យើងធ្វើការសាកល្បងលើការប្រើប្រាស់ប្រព័ន្ធថ្មីដើម្បីស្វែងរកនូវកំហុសឆ្គង និងបញ្ហាផ្សេងៗ ដែល អាចកើតឡើងដោយកំហុសបច្ចេកទេសណាមួយ ហើយធ្វើការកែប្រែ (Update) ទៅតាមជំហានដែល បានរៀបចំរចនាសម្ព័ន្ធខាងលើ នូវត្រង់ចំណុចណាដែលមានការខូចខាត (Error Code) ។ នៅក្នុងការ ធ្វើការសាកល្បងនេះ ត្រូវបានបែងចែកជាពីរប្រភេទគឺ៖

- Internal Test: គឺជាការសាកល្បងប្រើប្រាស់ ដោយធ្វើឡើងពីអ្នកសរសេរ Program ដើម្បី បង្កើតប្រព័ន្ធនេះឡើងពេលគឺ ក្រុមនិស្សិតក្នុងក្រុមទាំងអស់តែម្តង (Local Machine)។
- External Test: គឺជាការសាកល្បងធ្វើឡើងដោយ ដាក់ឲ្យអ្នកប្រើប្រាស់ធ្វើការសាកល្បងតែម្តង ។

២.២.៦ ការថែទាំ (Maintenance)

នៅក្នុងដំណាក់កាលនេះ បន្ទាប់ពីយើងបានដំឡើងប្រព័ន្ធឲ្យគេប្រើប្រាស់រួចរាល់អស់ហើយ យើងក៏នៅតែធ្វើការតាមដានដំណើរការ និងកង្វះខាតរបស់ប្រព័ន្ធ ហើយនិងធ្វើការជួសជុលប្រព័ន្ធ ប្រសិនបើប្រព័ន្ធមានបញ្ហា Error ចំណុចណាមួយកើតឡើង យើងនៅតែជួយ Support ទៅដល់អ្នកប្រើប្រាស់ ទៅតាមពេលវេលាដែលយើងអាចធ្វើបាន។

២.៣ វិធីសាស្ត្រប្រមូលទិន្នន័យ

រាល់ទិន្នន័យដែលក្រុមយើងខ្ញុំធ្វើការសិក្សាស្រាវជ្រាវសម្រាប់ការបង្កើតប្រព័ន្ធនេះគឺបានមកពីប្រភពចម្បង២ ៖

២.៣.១ ទិន្នន័យបឋម (Primary Data)

ជាទិន្នន័យដែលបានមកតាមរយៈការសាកសួរជាមួយ លោកគ្រូ អ្នកគ្រូ ដោយផ្ទាល់និង តាម រយៈ Online ដោយប្រើវិធីសាស្ត្ររៀបចំកម្រងសំណួរដើម្បីសាកសួរ ស្វែងយល់ពីតម្រូវការក៏ដូចជាព័ត៌មានមួយចំនួនដោយមានការឆ្លងកាត់យោបល់ពីសាស្ត្រ ចារ្យដឹកនាំ ដើម្បីទទួលបាននូវការកែតម្រូវ និងបច្ចេកទេសផ្សេងៗក្នុងការធ្វើឲ្យការប្រមូលទិន្នន័យ កាន់តែមានប្រសិទ្ធភាព។

២.៣.២ ទិន្នន័យបន្ទាប់មក (Secondary Data)

ជាទិន្នន័យដែលបានមកពី ការស្រាវជ្រាវទូទៅរបស់ក្រុមពួកយើង រួមមានទាំងការអាននូវនិក្ខេបបទ ស្នាដៃរបស់រៀបចំរួង របស់សាស្ត្រាចារ្យ ឯកសារពាក់ព័ន្ធនិងព័ត៌មានទូទៅដែលប្រមូលបាន តាមប្រព័ន្ធផ្សេងៗ ដូចជា Google, YouTube, Website,....។ល។

២.៤ គម្រោងពេលវេលានៃការស្រាវជ្រាវ

ក្នុងផ្នែកនេះផងដែរ គឺជាផ្នែកមួយដែលទាក់ទងទៅនឹងពេលវេលា ដែលប្រើសម្រាប់ការកំណត់ក្នុងការបង្កើតប្រព័ន្ធថ្មីនេះឡើង។ សម្រាប់គម្រោងពេលវេលានេះពិតជាមានសារៈសំខាន់ខ្លាំង ណាស់សម្រាប់យើងយកមកអនុវត្តក្នុងការកបង្កើតប្រព័ន្ធ ព្រោះថាគម្រោងពេលវេលានេះហើយ ដែលធ្វើឲ្យកិច្ចការស្រាវជ្រាវរបស់ពួកយើងប្រព្រឹត្តទៅបានយ៉ាងត្រឹមត្រូវតាមការកំណត់។ខាងក្រោមនេះគឺជាតារាងដែលត្រូវបានប្រើប្រាស់សម្រាប់កំណត់ពេលវេលាប្រព័ន្ធនេះឡើងរបស់ក្រុមនិស្សិតយើងខ្ញុំ៖

ល.រ	តារាងបែងចែកពេលវេលា	លទ្ធផលសម្រេចបាន
១	០៨/០៤/២០២៤ - ២០/០៥/២០២៤	ប្រមូលទិន្នន័យ
២	២១/០៥/២០២៤ – ០១/០៧/២០២៤	ដំណើរការសរសេរកូដ
៣	០២/០៧/២០២៤ – ១៥/០៨/២០២៤	សរសេរសារណា
៤	១៦/០៨/២០២៤ – ២៥/០៨/២០២៤	តេស្តសាកល្បង
៥	០៧/០៩/២០២៤	ប្រគល់ឯកសារ
៦	១០/១០/២០២៤	ការពារសារណា

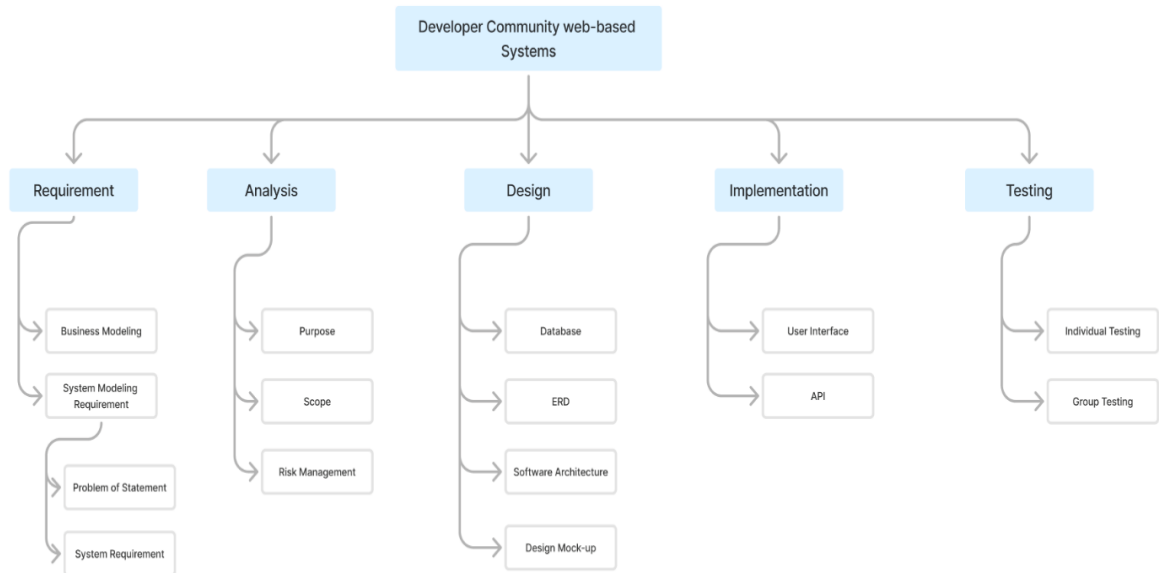
តារាងទី១ ៖ គម្រោងពេលវេលានៃការស្រាវជ្រាវ

ជំពូកទី៣

ការវិភាគទិន្នន័យ ការគ្រប់គ្រង និងការអនុវត្ត

៣.១ គម្រោងរបស់សម្ព័ន្ធនៃការតម្លើងប្រព័ន្ធ

៣.១.១ គម្រោងទិន្នន័យរបស់សម្ព័ន្ធ



រូបភាពទី១៥៖ Tree Structure View

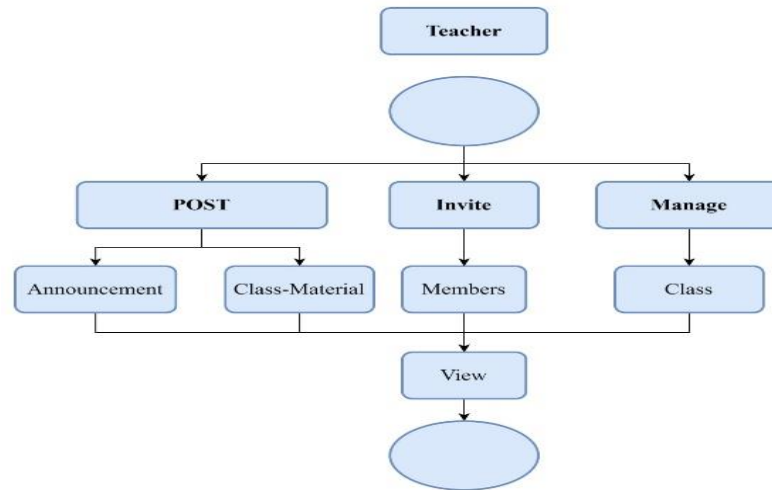
៣.២ ការតម្លើងប្រព័ន្ធគម្រោង

៣.២.១ តម្រូវការ (Requirement)

ដើម្បីសម្រេចបាននូវគោលដៅរបស់យើង យើងត្រូវរៀបចំផែនការមួយចំនួន។ ផ្នែកនេះនឹងរៀបរាប់លម្អិតអំពីអ្វីដែលកម្មវិធីរបស់យើងត្រូវការ រួមទាំងគំរូអាជីវកម្ម និងគំរូតម្រូវការប្រព័ន្ធ។

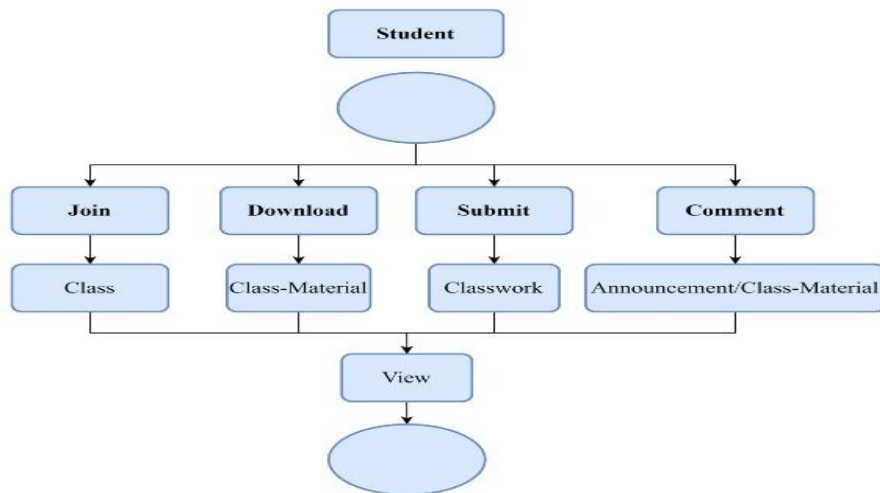
៣.២.១.១ តម្រូវការជីវិត (Business)

គំរូអាជីវកម្ម គឺជាក្របខ័ណ្ឌរួមមួយដើម្បីយល់ កំណត់ និងរចនាអាជីវកម្មនៅក្នុងទីផ្សារ ដែលជាវិធីដែលគម្រោងបង្កើតប្រាក់ចំណូល និងទទួលបានប្រាក់ចំណេញពីប្រតិបត្តិការគម្រោង។ កម្មវិធីដែលមានមូលដ្ឋានលើបណ្តាញរបស់យើងដែលហៅថា ការគ្រប់គ្រងប្រព័ន្ធថ្នាក់រៀនតាមបែបឌីជីថល ត្រូវបានបង្កើតឡើងដើម្បីគាំទ្រការគ្រប់គ្រងប្រព័ន្ធអប់រំសម្រាប់ លោកគ្រូ អ្នកគ្រូ និងសិស្សានុសិស្ស។



ដ្យាក្រាមទី១ ៖ សម្រាប់Teacher

- ❖ ដ្យាក្រាមសកម្មភាពនៃរបៀបដែលលោកគ្រូ អ្នកគ្រូ អាចធ្វើបានក្នុងថ្នាក់៖
 - ✓ លោកគ្រូ អ្នកគ្រូ អាចបង្កើតការប្រកាស ព្រឹត្តិការណ៍ សម្ភារៈថ្នាក់ និងកិច្ចការក្នុងថ្នាក់ ហើយក៏អាចជ្រើសរើសថ្នាក់ដើម្បីបង្ហាត់ផងដែរ។



ដ្យាក្រាម ទី២ ៖ សម្រាប់ Student

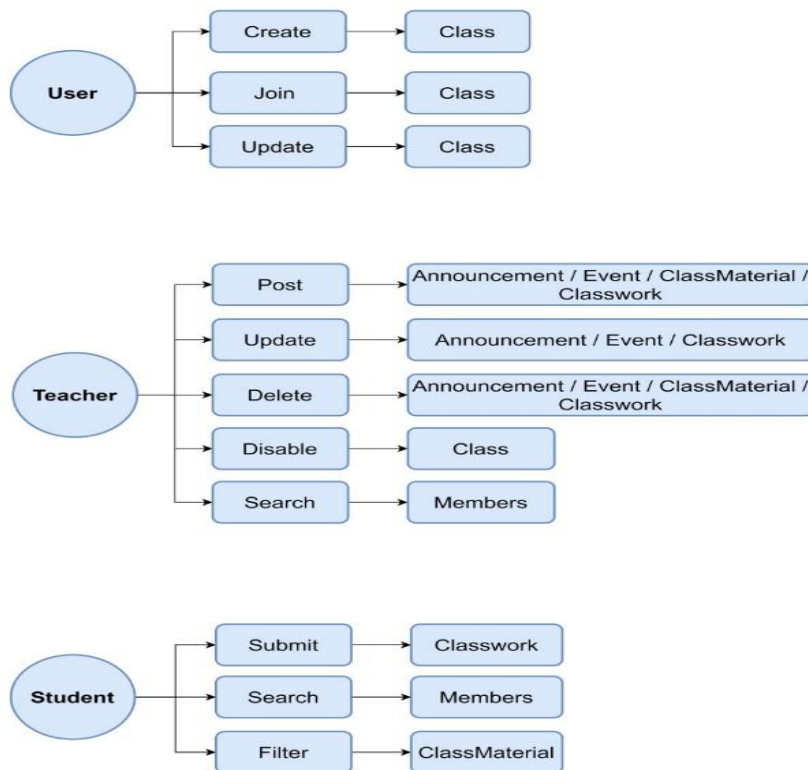
- ❖ ដ្យាក្រាមសកម្មភាពនៃរបៀបដែលសិស្សអាចធ្វើបាននៅក្នុងថ្នាក់៖
 - ✓ សិស្សអាចមើលការប្រកាស ព្រឹត្តិការណ៍ មតិយោបល់ និងទាញយកនៅក្នុងឯកសារក្នុងថ្នាក់ ហើយអាចបញ្ជូនកិច្ចការក្នុងថ្នាក់បាន។

៣.២.១.២ គំរូតម្រូវការប្រព័ន្ធ (System Requirement Modeling)

❖ បញ្ហានៃសេចក្តីថ្លែងការណ៍ (Problem of Statement) ៖

យោងតាមតម្រូវការដែលបានរៀបរាប់ខាងលើ អ្នករួមប្រើប្រាស់របស់យើងអាចធ្វើអ្វីដែលពួកគេត្រូវបានអនុញ្ញាតឱ្យធ្វើដូចជា បង្កើតថ្នាក់ ចូលរួមថ្នាក់ ធ្វើបច្ចុប្បន្នភាពថ្នាក់ អញ្ជើញសមាជិកឱ្យចូលរួមថ្នាក់ ប្រកាសប្រកាស ប្រកាសព្រឹត្តិការណ៍ មតិយោបល់ បង្កើតសម្ភារៈថ្នាក់ កិច្ចការថ្នាក់ ដាក់ស្នើ។ ការងារក្នុងថ្នាក់ និងមើលប្រវត្តិពិន្ទុទាំងអស់ដោយផ្អែកលើការអនុញ្ញាតរបស់ពួកគេ។ យ៉ាងណាក៏ដោយ នៅតែមានបញ្ហាមួយចំនួនទាក់ទងនឹងដំណើរការនៃការប្រើ ការគ្រប់គ្រងប្រព័ន្ធថ្នាក់រៀនតាមបែបឌីជីថល ។ ជាឧទាហរណ៍ គេហទំព័ររបស់យើងមិនអាចហៅជាវិដេអូ និងជជែកក្នុងថ្នាក់បានទេ ដូច្នេះហើយធ្វើឱ្យវាពិបាកសម្រាប់បុគ្គលម្នាក់ៗក្នុងការទំនាក់ទំនងបញ្ហារបស់ពួកគេទៅគ្នាទៅវិញទៅមកទាក់ទងនឹងការណែនាំតាមអ៊ីនធឺណិត។

❖ តម្រូវការប្រព័ន្ធ (System Requirement) ៖



ដ្យាក្រាម ទី៣ ៖ System Requirement

- ✓ គ្រូបង្រៀន៖ អាចបង្ហាញ និងលុបព្រឹត្តិការណ៍ប្រកាស សម្ភារៈថ្នាក់ និងការងារក្នុងថ្នាក់។ គ្រូក៏អាចធ្វើបច្ចុប្បន្នភាពការប្រកាស ព្រឹត្តិការណ៍ និងការងារក្នុងថ្នាក់ផងដែរ។ លើសពីនេះ គ្រូអាចបិទថ្នាក់ និងស្វែងរកសមាជិកក្នុងថ្នាក់ទាំងនោះ។
- ✓ សិស្ស៖ អាចបញ្ជូនស្នាដៃក្នុងថ្នាក់ ស្វែងរកសមាជិកក្នុងថ្នាក់របស់ពួកគេ។

៣.២.២ ការវិភាគប្រព័ន្ធ (Analysis)

៣.២.២.១ គោលបំណង (Purpose)

គោលបំណងស្នូលនៃការអភិវឌ្ឍន៍ ការគ្រប់គ្រងប្រព័ន្ធថ្នាក់រៀនតាមបែបឌីជីថល គឺផ្តល់ភាពងាយស្រួលក្នុងការគ្រប់គ្រងថ្នាក់រៀនតាមវិធីឌីជីថលសម្រាប់៖

❖ លោកគ្រូ អ្នកគ្រូ៖

- ✓ បង្កើតសម្ភារៈថ្នាក់ កិច្ចការ កម្រងសំណួរ ការប្រឡង។
- ✓ ផ្ទុកឡើងសម្ភារៈថ្នាក់។
- ✓ បង្កើតការប្រកាស។
- ✓ ការផ្តល់ពិន្ទុ និងការផ្តល់យោបល់លើការងាររបស់សិស្ស។

❖ សិស្សានុសិស្ស៖

- ✓ មើលកិច្ចការក្នុងថ្នាក់ ហើយបញ្ជូនវាទៅគ្រូរបស់ពួកគេ។
- ✓ មើលពិន្ទុ និងមតិកែលម្អពីគ្រូ។
- ✓ ទាញយកមេរៀន។
- ✓ ទទួលការជូនដំណឹងសម្រាប់ការអាប់ឡូតថ្មី។

៣.២.២.២ ផែនការវិសាលភាព (Scope)

ក្នុងនេះផងដែរ មានផែនការវិសាលភាពមួយចំនួនដែលយើងត្រូវបង្កើតកម្មវិធីគេហទំព័រនេះដូចជា៖

- ✓ គ្រប់គ្រងថ្នាក់រៀនដោយបង្កើតថ្នាក់ថ្មី ហើយអញ្ជើញគ្រូ និងសិស្សឱ្យចូលរួម។
- ✓ គ្រប់គ្រងសម្ភារក្នុងថ្នាក់ដោយការបង្ហាញ បញ្ចេញមតិ និងទាញយកសម្ភារៈ។
- ✓ គ្រប់គ្រងការប្រកាស និងការជូនដំណឹង។

៣.២.២.៣ ការគ្រប់គ្រងហានិភ័យ (Risk Management)

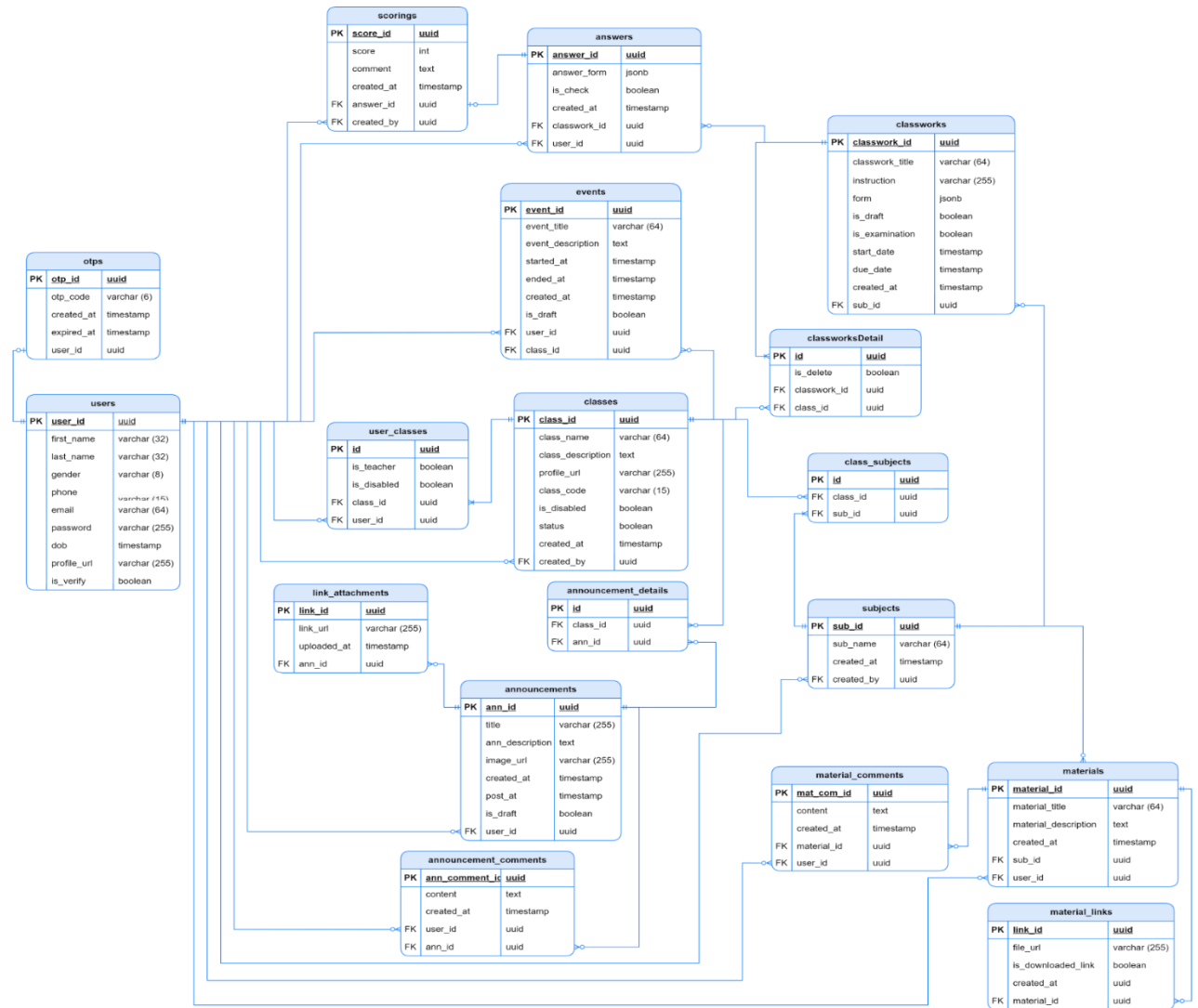
ការគ្រប់គ្រងហានិភ័យ គឺជាការកំណត់អត្តសញ្ញាណ ការវាយតម្លៃ និងការកំណត់អាទិភាពនៃហានិភ័យ ដោយការសម្របសម្រួល និងសន្សំសំចៃនៃធនធាន ដើម្បីកាត់បន្ថយ ការតាមដាន និងគ្រប់គ្រង ឬផលប៉ះពាល់នៃព្រឹត្តិការណ៍អកុសល ឬដើម្បីបង្កើនការសម្រេចនៃឱកាស។ ជាធម្មតា គម្រោងនីមួយៗតែងតែមានហានិភ័យ។ ដូច្នេះហើយ យើងត្រូវកំណត់អត្តសញ្ញាណ និងគ្រប់គ្រងហានិភ័យនៃគម្រោង ព្រោះវាជាផ្នែកសំខាន់មួយក្នុងការកសាងគម្រោង។

Risk	Description
Crash Project	Difficult to manage any changes in project requirement or scope.
Technical factors	User don't have experience with system.
Server Down	When many users' accesses at the same time.

៣.២.៣ ការរចនាប្រព័ន្ធ (Design)

៣.២.៣.១ ទំនាក់ទំនងទិន្នន័យ (Data Relationship)

ទំនាក់ទំនងទិន្នន័យ គឺជាផ្នែកមួយដែលបង្ហាញ អំពីទំនាក់ទំនង ទិន្នន័យរវាង Database និង Table ទាំងអស់។ Table Relationship ដែលបានបង្ហាញនៅក្នុង Diagram នេះសម្រាប់ជាគោល ដើម្បីកំណត់ឲ្យដឹងថា តើប្រព័ន្ធនេះមានទំនាក់ទំនងទិន្នន័យ និងលំហូរទិន្នន័យបែបណា។ ហើយចំណុចដែលសំខាន់សម្រាប់ Developer ក្នុងការអនុវត្តន៍ការងារសរសេរកូដរបស់ខ្លួន ជាពិសេសការប្រើប្រាស់ SQL Language ដើម្បីទំនាក់ទំនង Add, Access និង Process ទិន្នន័យនៅក្នុង Table ទាំងឡាយនៅក្នុង Database។



រូបភាពទី៤ ៖ Data Relationship

៣.២.៣.២ វចនានុក្រមទិន្នន័យ (Data Dictionary)

Data Dictionary or Data Repository គឺជាកន្លែងកណ្តាលសម្រាប់រក្សាព័ត៌មានអំពីទិន្នន័យប្រព័ន្ធ។ អ្នកវិភាគប្រើប្រាស់ data dictionary ដើម្បីប្រមូល កត់ត្រាទុក និងរៀបចំ facts ជាក់លាក់អំពី ប្រព័ន្ធដោយរួមបញ្ចូលទាំង contents នៃ data flows, data stores, external entities និង process ហើយ data dictionary ក៏កំណត់នឹងរៀបរាប់រាល់ data elements ទាំងអស់ព្រមទាំងការរួមបញ្ចូលគ្នារវាង data elements ដែលមានន័យ។

Table Name	Field Name	Datatype	Length	Required	PK/FK	Default	Reference	Description
otps	otp_id	uuid		Y	PK			Auto generate
	otp_code	varchar	6	Y				
	created_at	timestamp		Y				Timestamp without time zone
	expired_at	timestamp		Y				Timestamp without time zone
	user_id	uuid		Y	FK		users	
users	user_id	uuid		Y	PK			Auto generate
	first_name	varchar	32	Y				
	last_name	varchar	32	Y				
	gender	varchar	8					
	telephone	varchar	15					
	email	varchar	64	Y				
	password	varchar	255	Y				Timestamp without time zone
	dob	timestamp						unique
	is_verify	boolean		Y		0		1 = has verified 0 = not verified
	profile_url	varchar	255					
user_classes	id	serial		Y	PK			Auto generate
	is_teacher	boolean		Y			1	1 = teacher 0 = student
	is_disable	boolean		Y			0	1 = disabled 0 = enabled
	class_id	uuid		Y				
	user_id	uuid		Y	FK		classes	
classes	class_id	uuid		Y	FK		users	Auto generate
	class_name	varchar	64	Y	PK			
	class_description	text						
	profile_url	varchar	255					
	class_code	varchar	15	Y				
	is_disabled	boolean		Y		0		1 = disabled 0 = enabled
	status	boolean		Y		1		1 = active 0 = deleted
	created_at	timestamp		Y				Timestamp without time zone
announcement_details	created_by	uuid		Y	FK		users	
	id	uuid		Y	PK			Auto generate
	class_id	uuid		Y	FK		classes	
announcements	ann_id	uuid		Y	FK		announcements	
	title	varchar	255	Y				Auto generate
	ann_description	text		Y				
	created_at	timestamp		Y				Timestamp without time zone
	is_draft	boolean		Y		0		1 = draft 0 = post
link_attachments	user_id	uuid		Y	FK		users	
	link_id	uuid		Y	PK			Auto generate
link_attachments	link_url	varchar	255	Y			account to (user id)	

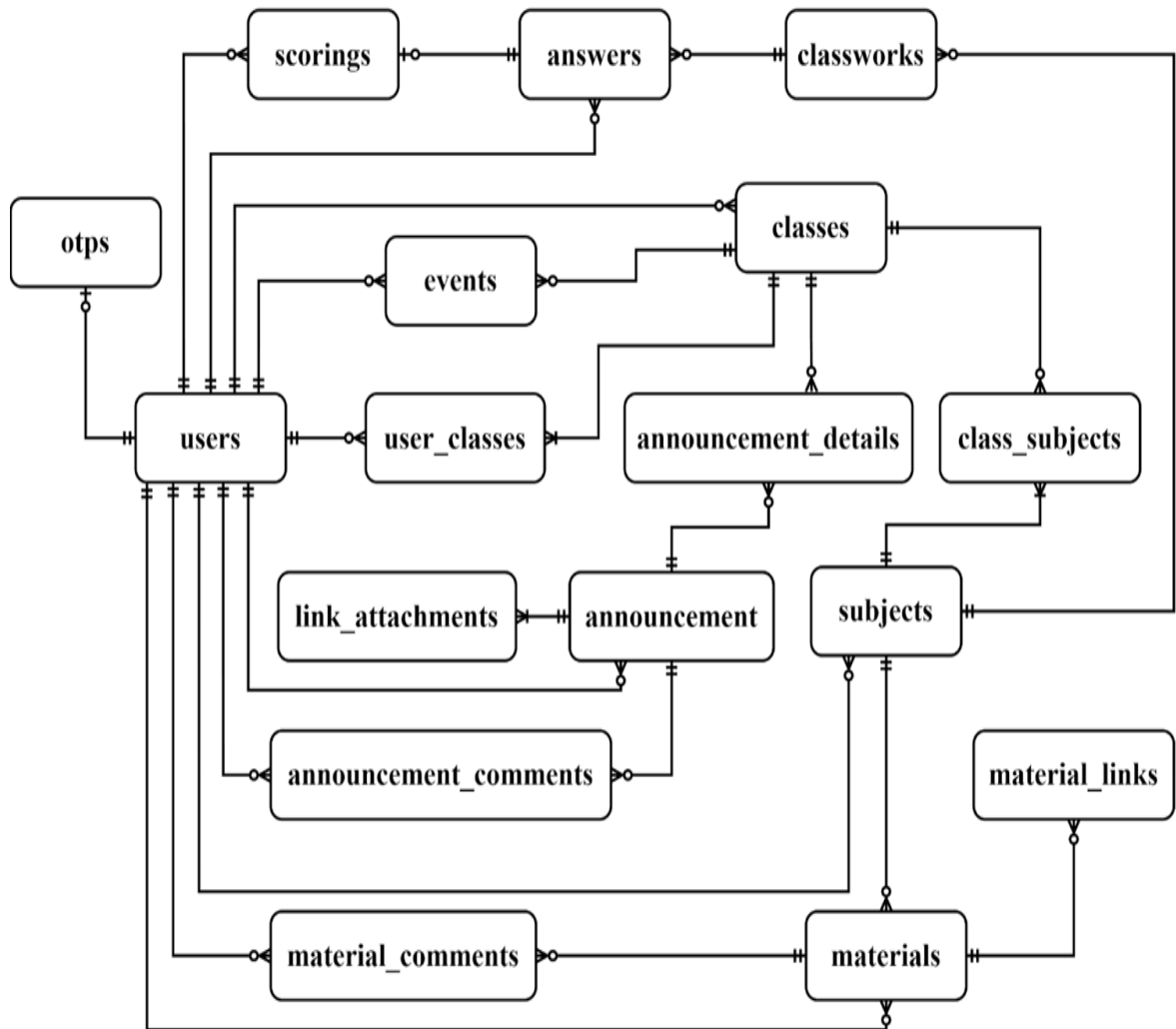
	uploaded_at	timestamp		Y			post_fb (post_id)	Timestamp without time zone
	ann_id	uuid		Y	PK	now		
announcement_comments	ann_comment_id	uuid		Y	PK			
	content	text		Y			account_fb (user_id)	
	created_at	timestamp		Y			post_fb (post_id)	
	user_id	uuid		Y	FK			
	ann_id	uuid		Y	FK			Timestamp without time zone
class_subjects	id	uuid		Y	PK			Auto generate
	class_id	uuid		Y	FK	classes		
	user_id	uuid		Y	FK	users		
subjects	sub_id	uuid		Y	PK			Auto generate
	sub_name	varchar	64	Y				
	created_at	timestamp		Y				Timestamp without time zone
	created_by	uuid		Y	FK		users	
materials	material_id	uuid		Y	PK			Auto generate
	material_title	varchar	64	Y				
	material_description	varchar	text					
	created_at	timestamp		Y				Timestamp without time zone
	sub_id	uuid		Y	FK		subjects	
	user_id	uuid		Y	FK		users	
material_comments	mat_com_id	uuid		Y	PK			Auto generate
	content	text		Y				
	created_at	timestamp						Timestamp without time zone
	material_id	uuid		Y	FK		materials	
	user_id	uuid		Y	FK		users	
	mat_com_id	uuid		Y	PK			Auto generate
material_links	link_id	uuid		Y	PK			Auto generate
	file_url	varchar	255					
	is_downloaded_link	boolean				0		1 = download link 0 = not download link
	created_at	timestamp						Timestamp without time zone
	material_id	uuid		Y	FK		materials	
events	event_id	uuid		Y	PK			Auto generate
	event_title	varchar	64	Y				
	event_description	text						
	started_at	timestamp		Y				Timestamp without time zone
	ended_at	timestamp		Y				Timestamp without time zone
	created_at	timestamp		Y				Timestamp without time zone
	is_draft	boolean		Y		0		1 = draft 0 = post
	user_id	uuid		Y	FK		users	
	class_id	uuid		Y	FK		classes	
classworks	classwork_id	uuid		Y	PK			Auto generate
	classwork_title	varchar	64	Y				

	instruction	varchar	255	Y				
	form	jsonb		Y				
	is_draft	boolean		Y		0		1 = draft 0 = post
	is_examination	boolean		Y		1		1 = exam 0 = assignment
	start_date	timestamp		Y				Timestamp without time zone
	due_date	timestamp		Y				Timestamp without time zone
	created_at	timestamp		Y				Timestamp without time zone
	sub_id	uuid		Y	FK		subjects	
answers	answer_id	uuid		Y	FK			Auto generate
	answer_form	jsonb		Y				
	is_check	boolean		Y		0		1 = has checked 0 = not checked
	created_at	timestamp		Y				Timestamp without time zone
	classwork_id	uuid		Y	FK		classworks	
	user_id	uuid		Y	FK		users	
scoreings	score_id	uuid		Y	FK			Auto generate
	score	numeric						Float number
	comment	varchar	255					
	created_at	timestamp		Y				Timestamp without time zone
	answer_id	uuid		Y	FK		answers	
	created_by	uuid		Y	FK		users	

តារាងទី២ ៖ Data Dictionary

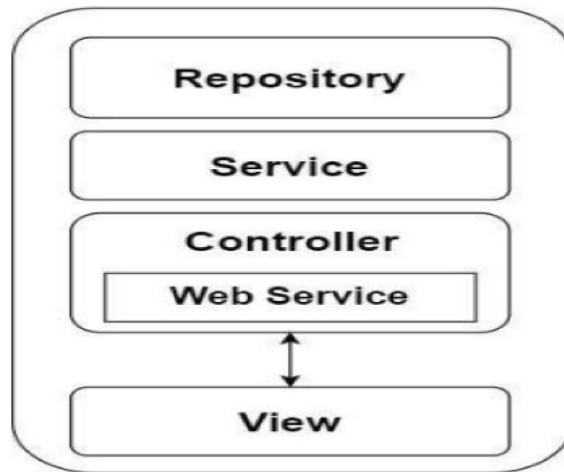
៣.២.៣.៣ ERD Diagram

An entity relationship diagram (ERD) គឺបង្ហាញពីទំនាក់ទំនងរវាងសំណុំអង្គភាពដែលរក្សាទុកក្នុងមូលដ្ឋានទិន្នន័យ។ អង្គភាពមួយនៅក្នុងបរិបទនេះគឺជាធាតុផ្សំនៃទិន្នន័យ។ នៅក្នុងពាក្យផ្សេងទៀត ERD បង្ហាញពីរចនាសម្ព័ន្ធទ្វេដងនៃមូលដ្ឋានទិន្នន័យ។



ដ្យាក្រាមទី៥ ៖ ERD Diagram

៣.២.៣.៤ Software Architecture



រូបភាពទី៦ ៖ Software Architecture

ដូចក្នុងរូបខាងលើ យើងសូមពណ៌នាអំពីលំហូរនៃ POJO៖

- ថ្នាក់ Web Service Rest Controller ហៅស្ថាបត្យកម្មកុំផ្លិចថ្នាក់សេវាកម្ម។
- ថ្នាក់សេវាកម្មមានអន្តរកម្មជាមួយប្រភពទិន្នន័យ (មូលដ្ឋានទិន្នន័យ) និងទាញយក ឬរៀបចំទិន្នន័យ។
- Web Service Rest Controller ឆ្លើយតបត្រឡប់ទៅកម្មវិធីអ្នកអាន។

បច្ចេកវិទ្យាដែលបានជ្រើសរើស៖

ម៉ាស៊ីនមេ(Server)៖

- POJO តំណាងឱ្យ Plain Old Java Object ដែលជាវត្ថុនាសម្ព័ន្ធទិន្នន័យសុទ្ធដែលមានវាលជាមួយ getters និង possibly setters ហើយអាចបដិសេធវិធីសាស្ត្រមួយចំនួនពី Object ឬ interface មួយចំនួនផ្សេងទៀតដូចជា Serializable ប៉ុន្តែមិនមានអាកប្បកិរិយារបស់វាទេ។

អតិថិជន(Client)៖

- Next.js៖ គឺជា React framework សម្រាប់បង្កើតកម្មវិធីគេហទំព័រពេញដង។ អ្នកប្រើ React Components ដើម្បីបង្កើតចំណុចប្រទាក់អ្នកប្រើ និង Next.js សម្រាប់មុខងារបន្ថែម និងការបង្កើនប្រសិទ្ធភាព។ វាផ្តល់នូវបរិយាកាសបង្ហាញពីរ ដែលជាផ្នែកខាងម៉ាស៊ីនភ្ញៀវ និងផ្នែកខាងម៉ាស៊ីនមេ។ HTML តំណាងឱ្យ HyperText Markup Language ។ វាជាសំណុំនៃស្លាកសម្គាល់ដែលប្រើសម្រាប់បង្កើតទំព័របណ្តាញ។

- Tailwind CSS៖ គឺជាកម្មវិធី CSS framework ដ៏មានប្រយោជន៍ដំបូងសម្រាប់បង្កើតគេហទំព័រ ទំនើបយ៉ាងឆាប់រហ័ស ដោយមិនចាំបាច់ចាក់ចេញពី HTML របស់អ្នក។
កម្មវិធីរបស់យើងគឺផ្អែកលើគំរូ MVC ដែលត្រូវបានអនុវត្តជាមួយ Spring Multi-Module ។ Spring Multi-Module មានធាតុផ្សំសំខាន់ៗចំនួនបួនដែលត្រូវប្រើក្នុងការអភិវឌ្ឍន៍កម្មវិធី៖
- A Repository៖ ថ្នាក់ឃ្លាំងបម្រើក្នុងស្រទាប់ជាប់លាប់នៃកម្មវិធីជា Data Access Object (DAO) ដែលមានតួនាទីជាចូលប្រើមូលដ្ឋានទិន្នន័យរបស់អ្នក។
សេវាកម្ម៖ តួនាទីអាជីវកម្មរបស់អ្នកទាំងអស់មាននៅក្នុងថ្នាក់សេវាកម្ម។
- ទិដ្ឋភាព៖ គឺជាបណ្តុំនៃថ្នាក់ដែលតំណាងឱ្យធាតុនៅក្នុងចំណុចប្រទាក់អ្នកប្រើប្រាស់ (អ្វីៗទាំងអស់ ដែលអ្នកប្រើប្រាស់អាចមើលឃើញ និងឆ្លើយតបនៅលើអេក្រង់ ដូចជាប៊ូតុង ការបង្ហាញ ប្រអប់។ល។)
ឧបករណ៍បញ្ជា៖ តំណាងឱ្យថ្នាក់ដែលភ្ជាប់សេវាកម្ម និងទិដ្ឋភាព ហើយត្រូវបានប្រើដើម្បីទំនាក់ទំនង រវាងថ្នាក់នៅក្នុងសេវាកម្ម និងទិដ្ឋភាព។

៣.២.៤ ការអភិវឌ្ឍន៍ (Implementation)

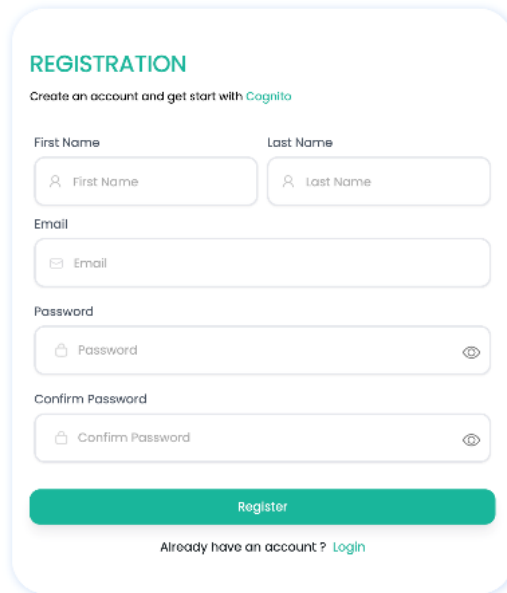
៣.២.៤.១ User Interface

- ❖ Login Form ប្រើសម្រាប់បង្ហាញពី ផ្ទាំងចូលប្រើប្រាស់

The image shows a login form titled "Login" with the subtitle "Welcome back to Cognito". It contains two input fields: "Email" with an envelope icon and "Password" with a lock icon and a toggle eye icon. Below these is a checkbox labeled "Remember me" and a red link "Forget password?". A large green "Login" button is centered below. Underneath the button is a link "Don't have an account ? Register". At the bottom is a button with the Google logo and the text "Login with Google".

រូបភាពទី១៦៖ Login Form

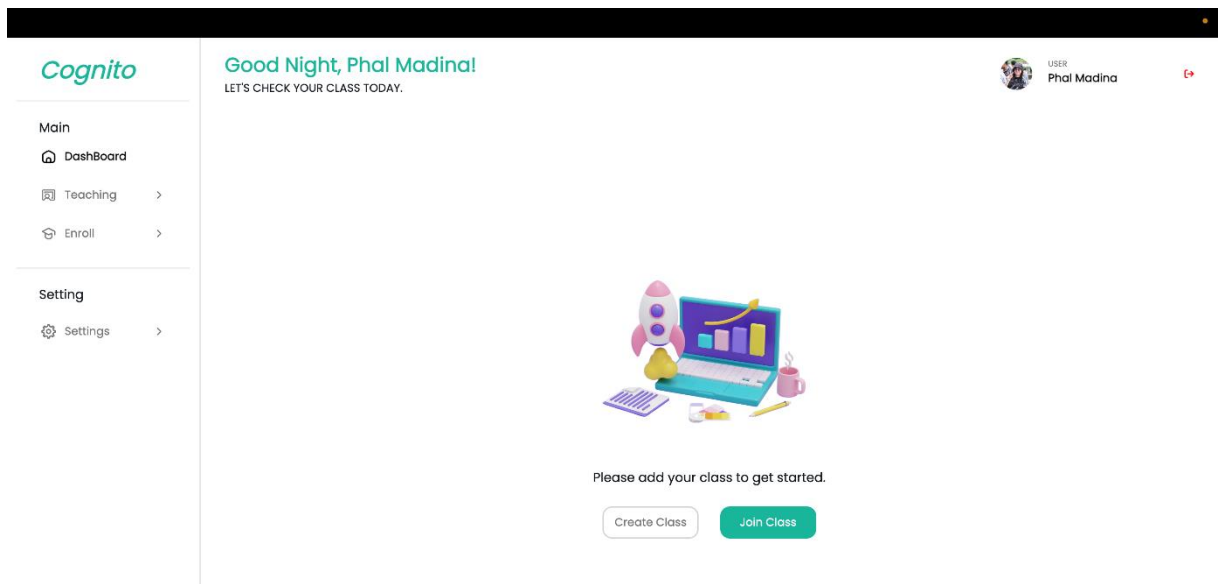
❖ Registration Form ប្រើសម្រាប់ចុះឈ្មោះចូលប្រើប្រាស់ប្រព័ន្ធ



The image shows a registration form titled "REGISTRATION" with the subtitle "Create an account and get start with Cognito". The form includes input fields for "First Name", "Last Name", "Email", "Password", and "Confirm Password". Each field has a small icon representing its content (e.g., a person icon for names, an envelope for email, and a key for passwords). There is a "Register" button at the bottom and a link "Already have an account ? Login" below it.

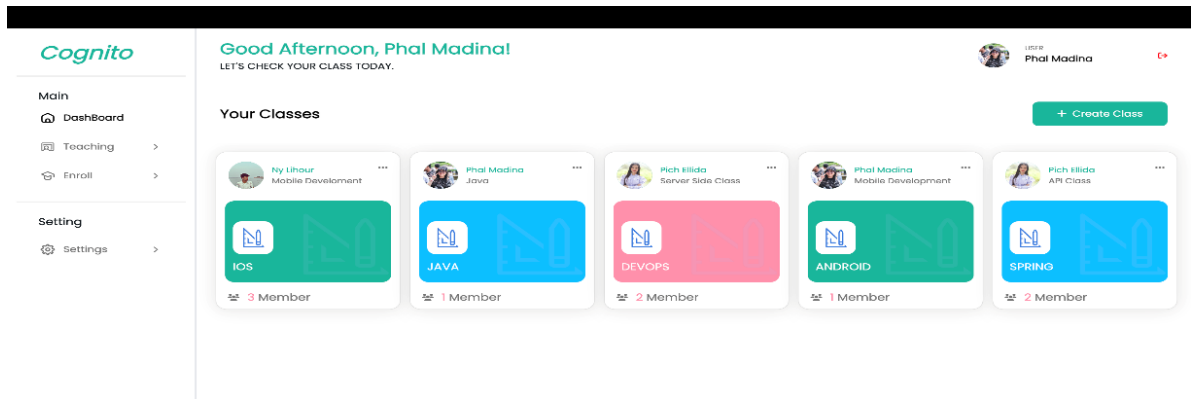
រូបភាពទី១៧៖Registration Form

❖ New user Homepage ប្រើសម្រាប់បង្ហាញផ្ទាំងទំព័រដើមសម្រាប់អ្នកប្រើប្រាស់ថ្មី



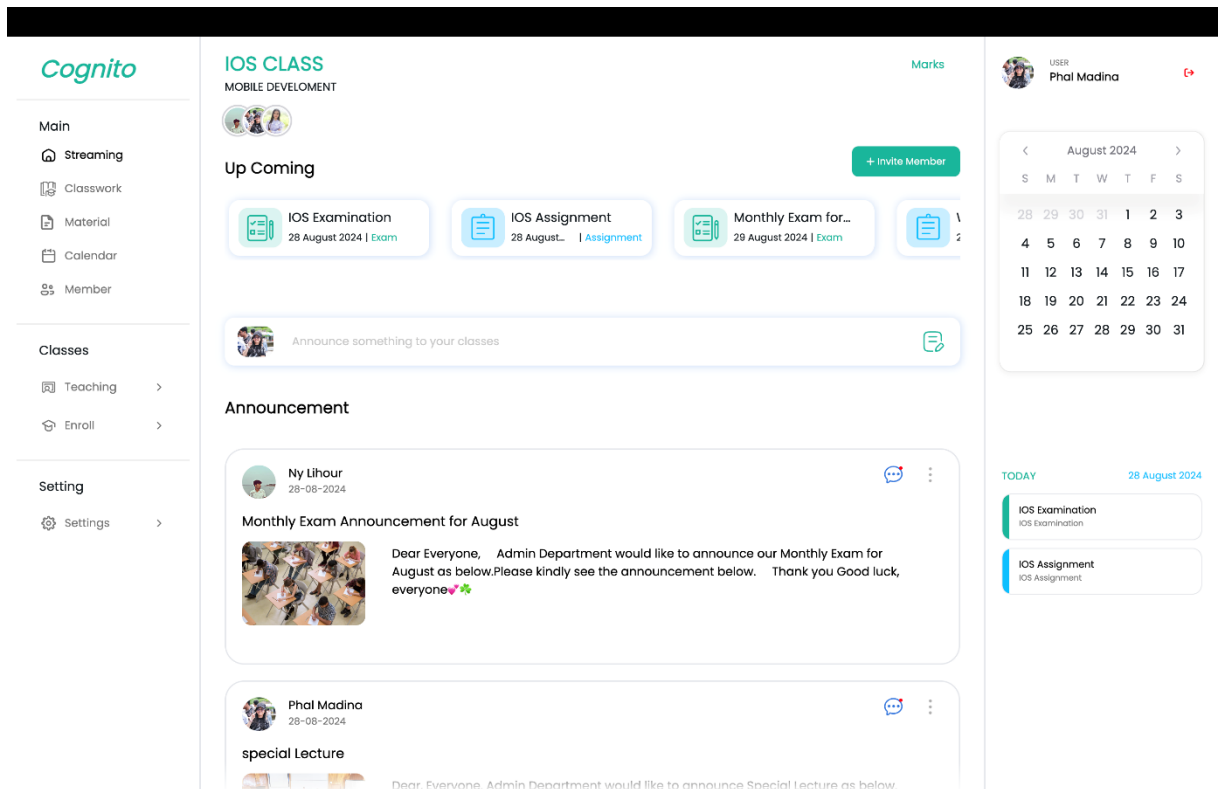
រូបភាពទី១៨៖New User Homepage

❖ After join class ផ្ទាំងបង្ហាញពីរូបភាពបន្ទាប់ពីចុះឈ្មោះចូលថ្នាក់រៀនហើយ



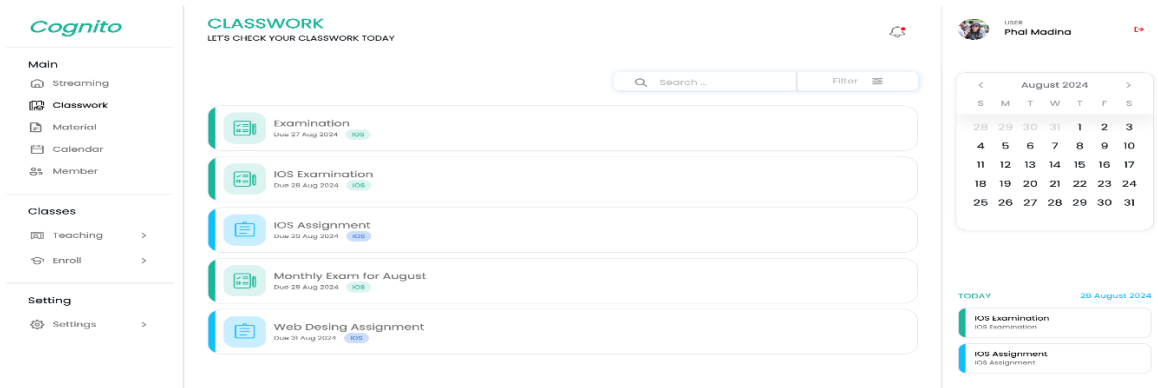
រូបភាពទី១៩៖ After Join Class

❖ Announcement Dashboard ប្រើសម្រាប់បង្ហាញព័ត៌មានក្នុងថ្នាក់



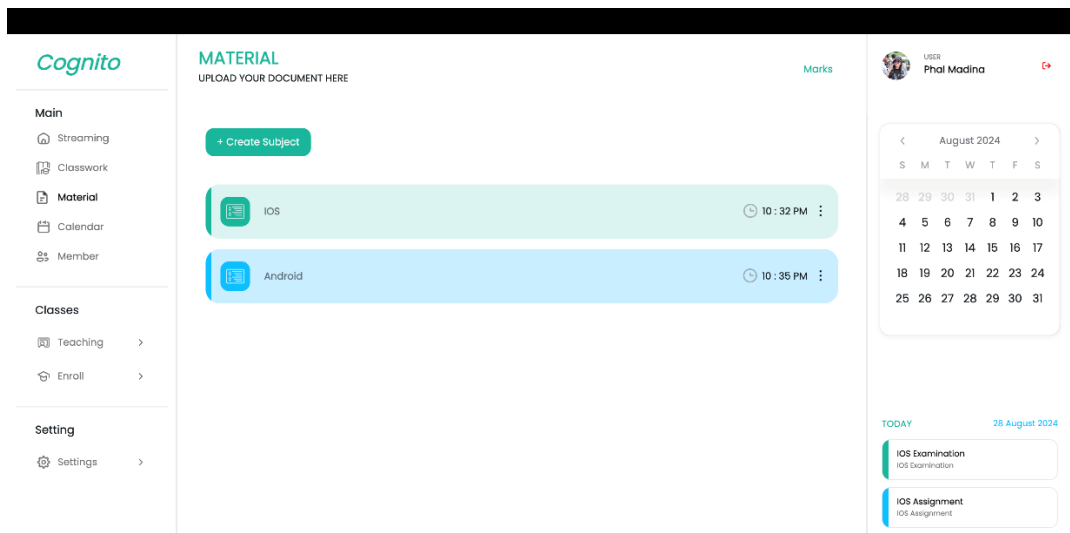
រូបភាពទី២០៖ Announcement Dashboard

❖ Classwork Dashboard ជាផ្ទាំងដើម្បីបង្ហាញការគ្រប់គ្រងការងារក្នុងថ្នាក់



រូបភាពទី២១៖ Classwork Dashboard

❖ Materials dashboard



រូបភាពទី២២៖ Materials Dashboard

❖ Member dashboard ផ្ទាំងបង្ហាញពីសមាជិកដែលបានចូលរួមក្នុងថ្នាក់

រូបភាពទី២៣៖ Member Dashboard

❖ User Setting – Setting

រូបភាពទី២៤៖ User Setting – Setting

❖ Class Setting– Setting

Cognito

Setting > Class

Class Details

Class Name
IOS

Description
Mobile Development

General

Invitation Code
For invitation code you can copy and sent you another member for join into your class.

Teacher : Th-MBIbZ6ya

Student : St-MBIbZ6ya

Disable classes
You can restore the classes that you had to disable.

phal madina
Java

Cancel save

រូបភាពទី២៥៖ Class Setting – Setting

សេចក្តីសន្និដ្ឋាន និង ការផ្តល់អនុសាសន៍

សេចក្តីសន្និដ្ឋាន

បន្ទាប់ពីក្រុមយើងខ្ញុំ បានសិក្សាស្រាវជ្រាវលើប្រធានបទ ដែលក្រុមយើងខ្ញុំបានធ្វើការសម្រេចចិត្ត បង្កើតនូវគេហទំព័រដែលល្អប្រសើរនេះរួចរាល់ ពួកយើងបានសង្កេតឃើញថា ប្រព័ន្ធគ្រប់គ្រងថ្នាក់រៀនតាមបែបឌីជីថលគឺជាឧបករណ៍សំខាន់ដែលអាចឱ្យលោកគ្រូ អ្នកគ្រូ គ្រប់គ្រងថ្នាក់រៀនរបស់ពួកគាត់កាន់តែមានប្រសិទ្ធភាពតាមរយៈការប្រើប្រាស់វិទ្យា។

ប្រព័ន្ធទាំងនេះជួយសម្រួលដល់កិច្ចការរដ្ឋបាលជាច្រើន ដូចជាការចូលរួម ការចាត់តាំង និងប្រមូលកិច្ចការផ្ទះ និងការតាមដានវឌ្ឍនភាពរបស់សិស្ស។ តាមរយៈការរៀបចំកិច្ចការទាំងនេះនៅក្នុងធីតាមួយ គ្រូបង្រៀនអាចសន្សំសំចៃពេលវេលា និងផ្ដោតការយកចិត្តទុកដាក់បន្ថែមទៀតលើការបង្រៀន និងការប្រាស្រ័យទាក់ទងជាមួយសិស្ស។

លើសពីនេះ ប្រព័ន្ធទាំងនេះ រួមបញ្ចូលមុខងារដូចជា សៀវភៅ និងឧបករណ៍ទំនាក់ទំនង ដែលជួយឱ្យសិស្ស និងមាតាបិតាជូនដំណឹងអំពីសកម្មភាពក្នុងថ្នាក់ និងថ្ងៃផុតកំណត់។

បន្ថែមពីលើការជួយសម្រួលដល់ការគ្រប់គ្រងថ្នាក់រៀន ប្រព័ន្ធឌីជីថលមានលក្ខណៈពិសេសនៅក្នុងបរិយាកាស ការសិក្សាពីចម្ងាយ។ ពួកគេអនុញ្ញាតឱ្យគ្រូបង្រៀនរក្សាចន្លោះម្តង និងការរៀបចំទោះបីជាសិស្សមិនមានវត្តមាននៅក្នុងថ្នាក់ក៏ដោយ។ ភាពស៊ីសង្វាក់គ្នានេះអាចជួយសិស្សឱ្យបន្តចូលរួម និងដើរលើផ្លូវជាមួយការរៀនសូត្ររបស់ពួកគេ ដោយមិនគិតពីទីតាំងរបស់ពួកគេ។

ការផ្តល់អនុសាសន៍

ក្រោយពីបានបញ្ចប់នូវការធ្វើប្រព័ន្ធនេះយើងឃើញថា ដើម្បីបង្កើនប្រសិទ្ធភាពនៃប្រព័ន្ធ គ្រប់គ្រងថ្នាក់រៀនតាមបែបឌីជីថល វាមានសារៈសំខាន់ណាស់ក្នុងការធ្វើតាមអនុសាសន៍សំខាន់ៗមួយ ចំនួន។

ទីមួយ សាលារៀនគួរតែវិនិយោគលើការបណ្តុះបណ្តាលដ៏ទូលំទូលាយសម្រាប់គ្រូបង្រៀន និង បុគ្គលិក។ ដើម្បីមានភាពសុំជាមួយលក្ខណៈពិសេស និងសមត្ថភាពរបស់ប្រព័ន្ធនឹងធ្វើឱ្យអ្នកអប់រំប្រើ ប្រាស់ឧបករណ៍នេះឱ្យអស់ពីសមត្ថភាព ដោយធានាថាសកម្មភាពក្នុងថ្នាក់រៀនត្រូវបានគ្រប់គ្រងយ៉ាង រលូន និងមានប្រសិទ្ធភាព។ បង្កើតវគ្គបណ្តុះបណ្តាល ឬសិក្ខាសាលាដែលធ្វើបច្ចុប្បន្នភាពជាទៀងទាត់ ក៏អាចជួយគ្រូឱ្យធ្វើបច្ចុប្បន្នភាពលើមុខងារថ្មីៗ និងការអនុវត្តល្អបំផុតផងដែរ។

ទីពីរ វាមានសារៈសំខាន់ណាស់ក្នុងការធានាឱ្យមានការចូលដំណើរការប្រកបដោយសមធម៌ ចំពោះបច្ចេកវិទ្យាចាំបាច់សម្រាប់សិស្សទាំងអស់។ នេះរួមបញ្ចូលទាំងការផ្តល់ឧបករណ៍ដូចជាកុំព្យូទ័រ យូរដៃ ឬថេប្លេត និងការភ្ជាប់អ៊ីនធឺណិតដែលអាចទុកចិត្តបាន ជាពិសេសសម្រាប់អ្នកនៅក្នុងសហគមន៍ក្រីក្រ។ សាលារៀនអាចពិចារណាលើការអនុវត្តកម្មវិធីកម្ចីសម្រាប់ឧបករណ៍ ឬសហការជាមួយអង្គការក្នុងស្រុកដើម្បីផ្តល់ជូននូវដំណោះស្រាយអ៊ីនធឺណិតដែលមានតម្លៃសមរម្យ។ ការធានាថាសិស្សទាំងអស់មានឧបករណ៍ដែលពួកគេត្រូវការ នឹងជួយសម្រួលដល់ការបែងចែកឌីជីថល និងធានាថាគ្មានសិស្សណាម្នាក់ត្រូវបានទុកចោល។

ជាចុងក្រោយគឺជាការជំរុញទំនាក់ទំនងបើកចំហជាមួយឪពុកម្តាយ និងសិស្សអំពីការប្រើប្រាស់ប្រព័ន្ធគ្រប់គ្រងថ្នាក់រៀនឌីជីថល។ គោលការណ៍ណែនាំ និងការរំពឹងទុកច្បាស់លាស់គួរតែត្រូវបានបង្កើតឡើង ដូច្នេះអ្នករាល់គ្នាយល់ពីរបៀបរុករកប្រព័ន្ធ និងប្រើប្រាស់លក្ខណៈពិសេសរបស់វាប្រកបដោយប្រសិទ្ធភាព។ ការផ្តល់ការបង្រៀន ឬធនធានជំនួយអាចផ្តល់សិទ្ធិអំណាចដល់សិស្ស និងមាតាបិតាឱ្យចូលរួមក្នុងវេទិកាប្រកបដោយទំនុកចិត្តជាងមុន។

ឯកសារយោង

1. Feature: www.classroom.google.com

www.google.com/forms/about/

www.edusync.com/google-sync/

2. User Interface: www.classroom.google.com
3. System Architecture: www.geeksforgeeks.org
4. <https://www.classvr.com/blog/what-is-a-digital-classroom/>
5. <https://digital.wings.uk.barclays/blog/the-digital-classroom-guide/>
6. <https://www.w3schools.com/>
7. <https://e.huawei.com/fr/knowledge/2024/industries/commercial-market/what-is-digital-classroom>
8. <https://www.lambdasolutions.net/blog/top-7-digital-classroom-tools-for-teachers-and-learners>
9. <https://www.waldenu.edu/programs/resource/seven-benefits-of-a-virtual-classroom>
10. 25 Aug 2024 <https://www.avanse.com/blog/benefits-of-creating-digital-classrooms-in-schools>
11. 21 Feb 2024 <https://corp.kaltura.com/blog/virtual-classroom-benefits/>

အိတ်ပေမ္မီ

➤ Login

```
1  "use client";
2  import React, { useEffect, useState } from "react";
3  import { Input, Checkbox, Button } from "@nextui-org/react";
4  import Link from "next/link";
5  import { signIn, signOut, useSession } from "next-auth/react";
6  import PasswordComponent from "../PasswordComponent";
7  import { useRouter } from "next/navigation";
8  import InputComponent from "../InputComponent";
9  import { useForm } from "react-hook-form";
10 import { authSchema } from "@/lib/schema/authSchema";
11 import { zodResolver } from "@hookform/resolvers/zod";
12 import toast, { Toaster } from "react-hot-toast";
13 import LoadingComponent from "../LoadingComponent";
14 import logo from "../../public/icon/logoFull.svg";
15 import Image from "next/image";
16 const LoginComponent = () => {
17   const router = useRouter();
18   const [loading, setLoading] = useState(false);
19   const {
20     register,
21     handleSubmit,
22     formState: { errors },
23   } = useForm({
24     resolver: zodResolver(authSchema),
25   });
26
27   async function handleLogin(loginInfo) {
28     setLoading(true);
29     const userLogin = {
30       email: loginInfo.email,
31       password: loginInfo.password,
32     };
33     const res = await signIn("credentials", {
34       redirect: false,
35       ...userLogin,
36     });
37     if (res.status === 200) {
38       toast.success("Login Successfully");
39       router.push("/dashboard");
40     } else {
41       setLoading(false);
42       toast.error("Please Checking your Email and Password again !");
43     }
44   }
45
46   const handleLoginWithGoogle = async () => {
47     const result = await signIn("google", { callbackUrl: "/dashBoard" });
48     toast.success("Login successfully");
49     setLoading(true);
50   };

```

```

1  return (
2    <div>
3      <div>
4        <Toaster position="top-center" reverseOrder={true} />
5      </div>
6      <div className="2xl:w-[500px] xl:w-[500px] lg:w-[500px] w-[91%] bg-white rounded-[40px] absolute 2xl:left-[48%]
7        xl:left-[35%] lg:left-[20%] left-[5%] 2xl:top-[50%] xl:top-[50%] lg:top-[50%] top-[50%] translate-y-[-55%] 2xl:py-[30px]
8        2xl:px-[40px] xl:py-[30px] xl:px-[40px] py-[40px] px-[20px] shadow-sd">
9        <div className="2xl:px-4 xl:px-4 flex">
10         <Image src={logo} alt="" className="2xl:w-[0px] 2xl:h-[0px] xl:w-[0px] xl:h-[0px] w-[45%]"></Image>
11       </div>
12       <div className="w-[100%] py-[7px]">
13         <h1 className="font-bold 2xl:text-[30px] xl:text-[28px] text-black text-[20px] mt-3">
14           Login
15         </h1>
16         <p className="2xl:text-[14px] xl:text-[13px] text-[#757575] py-1 text-[13px]">
17           Welcome back to Cognito
18         </p>
19       </div>
20       <form onSubmit={handleSubmit(handleLogin)}>
21         <div className="flex bg-white w-[100%] flex-wrap md:flex-nowrap gap-4 mt-[10px] mb-[10px]">
22           <div className="w-[100%] h-auto float-left">
23             <InputComponent
24               icon={
25                 <svg
26                   width="27"
27                   height="26"
28                   viewBox="0 0 27 26"
29                   fill="none"
30                   xmlns="http://www.w3.org/2000/svg"
31                 >
32                   <g opacity="0.3">
33                     <rect
34                       x="5.76367"
35                       y="7.5835"
36                       width="17.3333"
37                       height="13"
38                       rx="2"
39                       stroke="#33363F"
40                       strokeWidth="1.5"
41                     />
42                     <path
43                       d="M5.76367 10.8335L13.5359 14.7196C14.099 15.0011 14.7617 15.0011 15.3248 14.7196L23.097 10.8335"
44                       stroke="#33363F"
45                       strokeWidth="1.5"
46                     />
47                   </g>
48                 </svg>
49               }
50               valid={register}
51               name="email"
52               label="Email"
53               error={errors?.email}
54             />
55           </div>
56           <div>
57             <div className="w-[100%] mt-[7px] float-left">
58               <PasswordComponent
59                 valid={register}
60                 name="password"
61                 placeholder="Password"
62                 error={errors.password}
63               />
64             </div>
65             <Checkbox
66               className="mt-[8px] text-[13px]"
67               classNames={{
68                 label: "text-small",
69               }}
70             >
71               Remember me
72             </Checkbox>
73             <p className="text-small text-[red] float-right mt-[16px] text-[13px]">
74               <Link href="/forgotPassword"> Forget password ?</Link>
75             </p>

```

```

1  {loading ? (
2      <Button
3          type="submit"
4          color="primary"
5          variant="shadow"
6          className="w-[100%] mt-[10px]"
7      >
8          <LoadingComponent />
9      </Button>
10 ) : (
11     <Button
12         type="submit"
13         color="primary"
14         variant="shadow"
15         className="w-[100%] mt-[10px] text-[13px]"
16     >
17         Login
18     </Button>
19 ))
20 <p className="mt-[15px] text-center text-small text-[13px]">
21     Don't have an account ?{" "}
22     <Link href="/register" className="text-blue-400 text-[13px]">
23         Register
24     </Link>
25 </p>
26 <button
27     className="bg-white border py-2 w-full rounded-xl mt-5 flex justify-center items-center text-sm hover:scale-105 duration-300 hover:bg-[#dddffc]"
28     type="button"
29     onClick={handleLoginWithGoogle}
30 >
31     <svg
32         className="mr-3 text-[13px]"
33         xmlns="http://www.w3.org/2000/svg"
34         viewBox="0 0 48 48"
35         width="25px"
36     >
37         <path
38             fill="#FFC107"
39             d="M43.611,20.083H42V20H24v8h11.303c-1.649,4.657-6.08,8-11.303,8c-6.627,0-12-5.373-12-12c0-6.627,5.373-12,12-12c3.059,0,5.842,1.154,
40             7.961,3.039L5.657-5.657C34.046,6.053,29.268,4,24,4C12.955,4,4,12.955,4,24c0,11.045,8.955,20,20,20c11.045,0,20-8.955,20-20C44,22.659,
41             43.862,21.35,43.611,20.083z"
42         ></path>
43         <path
44             fill="#FF3D00"
45             d="M6.306,14.691L6.571,4.819C14.655,15.108,18.961,12,24,12c3.059,0,5.842,1.154,7.961,3.039L5.657-5.657C34.046,6.053,29.268,4,24,4C16.318,
46             4,9.656,8.337,6.306,14.691z"
47         ></path>
48         <path
49             fill="#4CAF50"
50             d="M24,44c5.166,0,9.86-1.977,13.409-5.192L6.19-5.238C29.211,35.091,26.715,36,24,36c-5.202,0-9.619-3.317-11.283-7.946L-6.522,5.025C9.505,
51             39.556,16.227,44,24,44z"
52         ></path>
53         <path
54             fill="#1976D2"
55             d="M43.611,20.083H42V20H24v8h11.303c0-7.92,2.237-2.231,4.166-4.087,5.51c0.001-0.001,0.002-0.001,0.003-0.002L6.19,5.238C36.971,39.205,44,
56             34,44,24C44,22.659,43.862,21.35,43.611,20.083z"
57         ></path>
58     </svg>
59     Login with Google
60 </button>
61 </form>
62 </div>
63 </div>
64 );
65 };
66
67 export default LoginComponent;
68

```

➤ Register



```
1 <div className="w-[100%] float-left p-[5px]">
2   <p className="text-gray-600 2xl:text-[14px] xl:text-[14px] text-[12px] xl:mb-[5px] 2xl:mb-[7px] mb-[5px]">
3     Email
4   </p>
5   <InputComponent
6     valid={register}
7     icon={
8       <svg
9         width="21"
10        height="20"
11        viewBox="0 0 27 26"
12        fill="none"
13        xmlns="http://www.w3.org/2000/svg"
14      >
15        <g opacity="0.3">
16          <rect
17            x="5.76367"
18            y="7.5835"
19            width="17.3333"
20            height="13"
21            rx="2"
22            stroke="#33363F"
23            strokeWidth="1.5"
24          />
25          <path
26            d="M5.76367 10.8335L13.5359 14.7196C14.099 15.0011 14.7617 15.0011 15.3248 14.7196L23.097 10.8335"
27            stroke="#33363F"
28            strokeWidth="1.5"
29          />
30        </g>
31      </svg>
32    }
33    name="email"
34    label="Email"
35    error={errors.email}
36  />
37 </div>
38 <div className="w-[100%] mt-[7px] float-left p-[5px]">
39   <p className="text-gray-600 2xl:text-[14px] xl:text-[14px] text-[12px] xl:mb-[5px] 2xl:mb-[7px] mb-[5px]">
40     Password
41   </p>
42   <PasswordComponent
43     valid={register}
44     name="password"
45     placeholder="Password"
46     error={errors.password}
47   />
48 </div>
```



```
1 <div className="w-[100%] mt-[7px] float-left p-[5px]">
2   <p className="text-gray-600 2xl:text-[14px] xl:text-[14px] text-[12px] xl:mb-[5px] 2xl:mb-[7px] mb-[5px]">
3     Confirm Password
4   </p>
5   <PasswordComponent
6     valid={register}
7     name="confirmPassword"
8     placeholder="Confirm Password"
9     error={errors.confirmPassword}
10  />
11 </div>
12 {loading ? (
13   <Button
14     type="submit"
15     variant="shadow"
16     className="w-[100%] mt-[25px] bg-[#387adf] text-white"
17   >
18     <LoadingComponent />
19   </Button>
20 ) : (
21   <Button
22     type="submit"
23     variant="shadow"
24     className="w-[100%] mt-[25px] bg-[#08b69b] text-white 2xl:text-[14px] xl:text-[14px] text-[12px] "
25   >
26     Register
27   </Button>
28 )}
29
30 <p className=" mt-[15px] text-center 2xl:text-[14px] xl:text-[14px] text-[12px] pb-[3%]">
31   Already have an account ? &nbsp;  
32   <Link href="/login" className="text-[#08b69b]">
33     Login
34   </Link>
35 </p>
36 </form>
37 );
38 };
39
40 export default RegisterComponent;
41
```

➤ Classwork

```
1 package org.example.classsphereapi.controller;
2
3 import io.swagger.v3.oas.annotations.Operation;
4 import io.swagger.v3.oas.annotations.security.SecurityRequirement;
5 import jakarta.validation.Valid;
6 import lombok.AllArgsConstructor;
7 import org.example.classsphereapi.model.dto.request.ClassworkRequest;
8 import org.example.classsphereapi.model.dto.response.ApiResponse;
9 import org.example.classsphereapi.model.dto.response.ClassworkResponse;
10 import org.example.classsphereapi.service.ClassworkService;
11 import org.springframework.http.HttpStatus;
12 import org.springframework.http.ResponseEntity;
13 import org.springframework.web.bind.annotation.*;
14 import java.time.LocalDateTime;
15 import java.util.List;
16 import java.util.UUID;
17
18 @RestController
19 @AllArgsConstructor
20 @RequestMapping("/api/v1/classwork")
21 @SecurityRequirement(name = "bearerAuth")
22 public class ClassworkController {
23     private final ClassworkService classworkService;
24     @GetMapping("/{classId}/classwork")
25     @Operation(summary = "Get All Classwork by Class-id")
26     public ResponseEntity<?> getAllClassworkByClassId(@PathVariable("classId") @Valid UUID classId){
27         List<ClassworkResponse> classworkResponse = classworkService.getAllClassworkByClassId(classId.toString());
28         ApiResponse<?> response = ApiResponse.builder()
29             .message("Classwork has get successfully")
30             .payload(classworkResponse)
31             .status(HttpStatus.OK)
32             .statusCode(HttpStatus.OK.value())
33             .time(LocalDateTime.now())
34             .build();
35         return ResponseEntity.status(HttpStatus.OK).body(response);
36     }
37     @GetMapping("/{Id}")
38     @Operation(summary = "Get All Classwork by id")
39     public ResponseEntity<?> getClassworkId(@PathVariable("Id") @Valid UUID classworkId){
40         ClassworkResponse classworkResponse = classworkService.getClassworkById(classworkId.toString());
41         ApiResponse<?> response = ApiResponse.builder()
42             .message("Classwork has get successfully")
43             .payload(classworkResponse)
44             .status(HttpStatus.OK)
45             .statusCode(HttpStatus.OK.value())
46             .time(LocalDateTime.now())
47             .build();
48         return ResponseEntity.status(HttpStatus.OK).body(response);
49     }
50     @GetMapping("/getAllClasswork/{subjectId}")
51     @Operation(summary = "Get All Classwork by subject id")
52     public ResponseEntity<?> getAllClassworkBySubjectId(@PathVariable("subjectId") @Valid UUID subjectId){
53         List<ClassworkResponse> classworkResponse = classworkService.getAllClassworkBySubjectId(subjectId.toString());
54         ApiResponse<?> response = ApiResponse.builder()
55             .message("Classwork created successfully")
56             .payload(classworkResponse)
57             .status(HttpStatus.OK)
58             .statusCode(HttpStatus.OK.value())
59             .time(LocalDateTime.now())
60             .build();
61         return ResponseEntity.status(HttpStatus.OK).body(response);
62     }
63 }
```

```

1  @PostMapping("/postClasswork/{subjectId}")
2  @Operation(summary = "Post classwork")
3  public ResponseEntity<?> createExamination(@PathVariable("subjectId") @Valid UUID subjectId, @RequestBody @Valid ClassworkRequest classworkRequest){
4      ClassworkResponse classworkResponse = classworkService.createClasswork(subjectId.toString(), classworkRequest);
5      ApiResponse<?> response = ApiResponse.builder()
6          .message("Classwork get successfully")
7          .payload(classworkResponse)
8          .status(HttpStatus.CREATED)
9          .statusCode(HttpStatus.CREATED.value())
10         .time(LocalDateTime.now())
11         .build();
12     return ResponseEntity.status(HttpStatus.CREATED).body(response);
13 }
14
15 @PutMapping("/updateClasswork/{classworkId}")
16 @Operation(summary = "Update All Classwork by id")
17 public ResponseEntity<?> updateExamination(@PathVariable("classworkId") @Valid UUID classwork, @RequestBody ClassworkRequest classworkRequest){
18     ClassworkResponse classworkResponse = classworkService.updateClasswork(classwork.toString(), classworkRequest);
19     ApiResponse<?> response = ApiResponse.builder()
20         .message("Classwork updated successfully")
21         .payload(classworkResponse)
22         .status(HttpStatus.OK)
23         .statusCode(HttpStatus.OK.value())
24         .time(LocalDateTime.now())
25         .build();
26     return ResponseEntity.status(HttpStatus.OK).body(response);
27 }
28
29 @DeleteMapping("/deleteClasswork")
30 @Operation(summary = "Delete Classwork by id")
31 public ResponseEntity<?> deleteClasswork(@RequestParam String classworkId){
32     classworkService.deleteClasswork(classworkId);
33     ApiResponse<?> response = ApiResponse.builder()
34         .message("Classwork updated successfully")
35         .status(HttpStatus.OK)
36         .statusCode(HttpStatus.OK.value())
37         .time(LocalDateTime.now())
38         .build();
39     return ResponseEntity.status(HttpStatus.OK).body(response);
40 }
41
42 @DeleteMapping("/deleteClasswork/{class-id}")
43 @Operation(summary = "Delete classwork from class")
44 public ResponseEntity<?> deleteExaminationFromClass(@PathVariable("class-id") UUID classId, @RequestParam String classworkId){
45     classworkService.deleteClassworkFromClass(classId.toString(), classworkId);
46     ApiResponse<?> response = ApiResponse.builder()
47         .message("Classwork updated successfully")
48         .status(HttpStatus.OK)
49         .statusCode(HttpStatus.OK.value())
50         .time(LocalDateTime.now())
51         .build();
52     return ResponseEntity.status(HttpStatus.OK).body(response);
53 }
54 }
55

```

➤ Member

```
1 package org.example.classspHEREapi.controller;
2
3 import io.swagger.v3.oas.annotations.security.SecurityRequirement;
4 import jakarta.validation.Valid;
5 import jakarta.validation.constraints.Email;
6 import lombok.AllArgsConstructor;
7 import org.example.classspHEREapi.model.dto.response.ApiResponse;
8 import org.example.classspHEREapi.model.dto.response.ClassMemberResponse;
9 import org.example.classspHEREapi.service.ClassMemberService;
10 import org.springframework.http.HttpStatus;
11 import org.springframework.http.ResponseEntity;
12 import org.springframework.web.bind.annotation.*;
13 import java.time.LocalDateTime;
14 import java.util.UUID;
15
16 @RestController
17 @AllArgsConstructor
18 @RequestMapping("/api/v1/class-member")
19 @SecurityRequirement(name = "bearerAuth")
20 public class ClassMemberController {
21     private final ClassMemberService classMemberService;
22     @GetMapping("/{getAllMemberInClass/{classId}")
23     public ResponseEntity<?> getAllMemberInClass(@PathVariable String classId){
24         ApiResponse<?> response = ApiResponse.builder()
25             .message("Get all members has been successfully.")
26             .payload(classMemberService.getAllMemberInClass(classId))
27             .status(HttpStatus.CREATED)
28             .statusCode(HttpStatus.CREATED.value())
29             .time(LocalDateTime.now())
30             .build();
31         return ResponseEntity.status(HttpStatus.OK).body(response);
32     }
33     @PutMapping("/changeRole/{classId}/{userId}")
34     public ResponseEntity<?> updateMembersRole(@PathVariable("classId") @Valid UUID classId, @Valid @PathVariable("userId") UUID userId, @RequestParam(defaultValue = "false") boolean role ){
35         ClassMemberResponse classMemberResponse=classMemberService.updateRole(classId.toString(),userId.toString(),role);
36         ApiResponse<?> response = ApiResponse.builder()
37             .message("Update members role has been successfully.")
38             .payload(classMemberResponse)
39             .status(HttpStatus.OK)
40             .statusCode(HttpStatus.OK.value())
41             .time(LocalDateTime.now())
42             .build();
43         return ResponseEntity.status(HttpStatus.OK).body(response);
44     }
45
46     @DeleteMapping("/delete-member-from-class/{classId}")
47     public ResponseEntity<?> deleteMemberFromClass(@PathVariable @Valid UUID classId,@RequestParam @Valid UUID userId){
48         classMemberService.removeMember(classId.toString(),userId.toString());
49         ApiResponse<?> response = ApiResponse.builder()
50             .message("Remove members form class has been successfully.")
51             .status(HttpStatus.OK)
52             .statusCode(HttpStatus.OK.value())
53             .time(LocalDateTime.now())
54             .build();
55         return ResponseEntity.status(HttpStatus.OK).body(response);
56     }
57
58     @PostMapping("/joinClassByCode/{classCode}")
59     public ResponseEntity<?> joinClassByCode(@PathVariable String classCode){
60         ApiResponse<?> response = ApiResponse.builder()
61             .message("Join class by code has been successfully.")
62             .payload(classMemberService.joinClassByCode(classCode))
63             .status(HttpStatus.OK)
64             .statusCode(HttpStatus.CREATED.value())
65             .time(LocalDateTime.now())
66             .build();
67         return ResponseEntity.status(HttpStatus.OK).body(response);
68     }
69
70     @PostMapping("/joinClass/{classId}/{email}")
71     public ResponseEntity<?> joinClassByEmail(@PathVariable @Valid @Email String email, @PathVariable("classId") @Valid UUID classId, @RequestParam(defaultValue = "false") Boolean isTeacher){
72         ApiResponse<?> response = ApiResponse.builder()
73             .message("Join class by email successfully.")
74             .payload(classMemberService.joinClassByEmail(email,classId.toString(),isTeacher))
75             .status(HttpStatus.OK)
76             .statusCode(HttpStatus.CREATED.value())
77             .time(LocalDateTime.now())
78             .build();
79         return ResponseEntity.status(HttpStatus.OK).body(response);
80     }
81 }
82 }
83 }
```

➤ Material

```
1 package org.example.classsphereapi.controller;
2
3 import io.swagger.v3.oas.annotations.Operation;
4 import io.swagger.v3.oas.annotations.security.SecurityRequirement;
5 import jakarta.validation.Valid;
6 import lombok.AllArgsConstructor;
7 import org.example.classsphereapi.model.dto.request.MaterialsRequest;
8 import org.example.classsphereapi.model.dto.response.ApiResponse;
9 import org.example.classsphereapi.service.MaterialsService;
10 import org.springframework.http.HttpStatus;
11 import org.springframework.http.ResponseEntity;
12 import org.springframework.web.bind.annotation.*;
13 import java.time.LocalDateTime;
14 import java.util.UUID;
15
16 @RestController
17 @AllArgsConstructor
18 @RequestMapping("/api/v1/materials")
19 @SecurityRequirement(name = "bearerAuth")
20 public class MaterialController {
21     private final MaterialsService materialsService;
22     @GetMapping("/{getAllMaterials/{subjectId}")
23     @Operation(summary = "get All Materials By subjectId ")
24     public ResponseEntity<> getAllMaterialsBySubjectId(@PathVariable("subjectId") @Valid UUID subjectId) {
25         ApiResponse<> response = ApiResponse.builder()
26             .message("Material has been gotten successfully.")
27             .payload(materialsService.getAllMaterialsBySubjectId(subjectId.toString()))
28             .status(HttpStatus.OK)
29             .statusCode(HttpStatus.OK.value())
30             .time(LocalDateTime.now())
31             .build();
32         return ResponseEntity.status(HttpStatus.OK).body(response);
33     }
34
35     @GetMapping("/getAllMaterials")
36     @Operation(summary = "For get All Material ")
37     public ResponseEntity<> getAllMaterials() {
38         ApiResponse<> response = ApiResponse.builder()
39             .message("Material has been gotten successfully.")
40             .payload(materialsService.getAllMaterials())
41             .status(HttpStatus.OK)
42             .statusCode(HttpStatus.OK.value())
43             .time(LocalDateTime.now())
44             .build();
45         return ResponseEntity.status(HttpStatus.OK).body(response);
46     }
47
48     @GetMapping("/getMaterialById/{materialId}")
49     @Operation(summary = "For get All Material By Id")
50     public ResponseEntity<> getMaterialById(@PathVariable("materialId") @Valid UUID materialId) {
51         ApiResponse<> response = ApiResponse.builder()
52             .message("Get material by id successfully")
53             .payload(materialsService.getMaterialById(materialId.toString()))
54             .status(HttpStatus.OK)
55             .statusCode(HttpStatus.OK.value())
56             .time(LocalDateTime.now())
57             .build();
58         return ResponseEntity.status(HttpStatus.OK).body(response);
59     }
60
61     @PostMapping("/createMaterial/{subjectId}")
62     @Operation(summary = "For Create Material")
63     public ResponseEntity<> createMaterial(@PathVariable("subjectId") UUID subjectId, @RequestBody @Valid MaterialsRequest materialRequest) {
64         ApiResponse<> response = ApiResponse.builder()
65             .message("Material has been created successfully.")
66             .payload(materialsService.createMaterial(materialRequest, subjectId.toString()))
67             .status(HttpStatus.CREATED)
68             .statusCode(HttpStatus.CREATED.value())
69             .time(LocalDateTime.now())
70             .build();
71         return ResponseEntity.status(HttpStatus.CREATED).body(response);
72     }
73
74     @PutMapping("/updateMaterialById/{materialId}/{subjectId}")
75     @Operation(summary = "For Update Material Comment")
76     public ResponseEntity<> updateMaterialById(@PathVariable @Valid UUID materialId, @PathVariable("subjectId") @Valid UUID subjectId, @RequestBody @Valid MaterialsRequest materialRequest) {
77         ApiResponse<> response = ApiResponse.builder()
78             .message("Update Material successfully")
79             .payload(materialsService.updateMaterialById(materialId.toString(), subjectId.toString(), materialRequest))
80             .status(HttpStatus.OK)
81             .statusCode(HttpStatus.OK.value())
82             .time(LocalDateTime.now())
83             .build();
84         return ResponseEntity.status(HttpStatus.OK).body(response);
85     }
86
87     @DeleteMapping("/deleteMaterialById/{materialId}")
88     @Operation(summary = "Delete Materials By Id")
89     public ResponseEntity<> deleteMaterialById(@PathVariable @Valid UUID materialId) {
90         materialsService.deleteMaterialById(materialId.toString());
91         ApiResponse<> response = ApiResponse.builder()
92             .message("Delete Material with id " + materialId + " successfully")
93             .status(HttpStatus.OK)
94             .statusCode(HttpStatus.OK.value())
95             .time(LocalDateTime.now())
96             .build();
97         return ResponseEntity.status(HttpStatus.OK).body(response);
98     }
99 }
```

➤ Class

```
1 package org.example.classsphereapi.controller;
2
3
4 import io.swagger.v3.oas.annotations.Operation;
5 import io.swagger.v3.oas.annotations.security.SecurityRequirement;
6 import jakarta.validation.Valid;
7 import lombok.AllArgsConstructor;
8 import org.example.classsphereapi.model.dto.request.ClassRequest;
9 import org.example.classsphereapi.model.dto.response.ApiResponse;
10 import org.example.classsphereapi.service.ClassService;
11 import org.hibernate.validator.constraints.UUID;
12 import org.springframework.http.HttpStatus;
13 import org.springframework.http.ResponseEntity;
14 import org.springframework.web.bind.annotation.*;
15 import java.time.LocalDateTime;
16
17 @RestController
18 @AllArgsConstructor
19 @RequestMapping("/api/v1/classes")
20 @SecurityRequirement(name = "bearerAuth")
21 public class ClassController {
22     private final ClassService classService;
23
24     @GetMapping
25     @Operation(summary = "Get all classes")
26     public ResponseEntity<?> getAllClass(){
27         ApiResponse<?> response = ApiResponse.builder()
28             .message("Get all classes successfully.")
29             .payload(classService.getAllClass())
30             .time(LocalDateTime.now())
31             .status(HttpStatus.OK)
32             .statusCode(HttpStatus.OK.value())
33             .build();
34         return ResponseEntity.status(HttpStatus.OK).body(response);
35     }
36
37     @GetMapping("/{classId}")
38     @Operation(summary = "Get class by id")
39     public ResponseEntity<?> getClassById(@PathVariable @UUID String classId){
40         ApiResponse<?> response = ApiResponse.builder()
41             .message("Get classes by id successfully.")
42             .payload(classService.getClassById(classId))
43             .status(HttpStatus.OK)
44             .statusCode(HttpStatus.OK.value())
45             .time(LocalDateTime.now())
46             .build();
47         return ResponseEntity.status(HttpStatus.OK).body(response);
48     }
49
50     @PostMapping
51     @Operation(summary = "Create class")
52     public ResponseEntity<?> createClass(@RequestBody @Valid ClassRequest classRequest){
53         ApiResponse<?> response = ApiResponse.builder()
54             .message("Class has been created successfully.")
55             .payload(classService.createClass(classRequest))
56             .status(HttpStatus.CREATED)
57             .statusCode(HttpStatus.CREATED.value())
58             .time(LocalDateTime.now())
59             .build();
60         return ResponseEntity.status(HttpStatus.CREATED).body(response);
61     }
62
63     @PutMapping("/{classId}")
64     @Operation(summary = "Update class by id")
65     public ResponseEntity<?> updateClassById(@PathVariable @UUID String classId, @RequestBody @Valid ClassRequest classRequest){
66         ApiResponse<?> response = ApiResponse.builder()
67             .message("Class has been updated successfully.")
68             .payload(classService.updateClassById(classId, classRequest))
69             .status(HttpStatus.OK)
70             .statusCode(HttpStatus.OK.value())
71             .time(LocalDateTime.now())
72             .build();
73         return ResponseEntity.status(HttpStatus.OK).body(response);
74     }
75
76     @PutMapping("/disabled-class/{classId}")
77     @Operation(summary = "Disabled class by id")
78     public ResponseEntity<?> disabledClassById(@PathVariable @UUID String classId, Boolean status){
79         ApiResponse<?> response = ApiResponse.builder()
80             .message("Class has been disabled successfully.")
81             .payload(classService.disabledClassById(classId, status))
82             .status(HttpStatus.OK)
83             .statusCode(HttpStatus.OK.value())
84             .time(LocalDateTime.now())
85             .build();
86         return ResponseEntity.status(HttpStatus.OK).body(response);
87     }
88 }
89
```

➤ Auth

```
1 package org.example.classssphereapi.controller;
2
3
4 import io.swagger.v3.oas.annotations.Operation;
5 import io.swagger.v3.oas.annotations.security.SecurityRequirement;
6 import jakarta.validation.Valid;
7 import lombok.AllArgsConstructor;
8 import org.example.classssphereapi.exception.NotFoundException;
9 import org.example.classssphereapi.jwt.JwtService;
10 import org.example.classssphereapi.model.dto.request.AuthRequest;
11 import org.example.classssphereapi.model.dto.request.ForgetPasswordRequest;
12 import org.example.classssphereapi.model.dto.request.UserRequest;
13 import org.example.classssphereapi.model.dto.response.ApiResponse;
14 import org.example.classssphereapi.model.dto.response.AuthResponse;
15 import org.example.classssphereapi.model.dto.response.UserResponse;
16 import org.example.classssphereapi.service.AuthService;
17 import org.modelmapper.ModelMapper;
18 import org.springframework.http.HttpStatus;
19 import org.springframework.http.ResponseEntity;
20 import org.springframework.security.authentication.AuthenticationManager;
21 import org.springframework.security.authentication.BadCredentialsException;
22 import org.springframework.security.authentication.DisabledException;
23 import org.springframework.security.authentication.UsernamePasswordAuthenticationToken;
24 import org.springframework.security.core.userdetails.UserDetails;
25 import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
26 import org.springframework.web.bind.annotation.*;
27 import java.time.LocalDateTime;
28
29 @RestController
30 @AllArgsConstructor
31 @RequestMapping("/api/v1/auth")
32 public class AuthController {
33     private final AuthService authService;
34     private final BCryptPasswordEncoder passwordEncoder;
35     private final AuthenticationManager authenticationManager;
36     private final ModelMapper modelMapper;
37     private final JwtService jwtService;
38     private void authenticate(String username, String password) throws Exception {
39         try {
40             UserDetails userApp = authService.loadUserByUsername(username);
41             if (userApp == null) {
42                 throw new NotFoundException("Wrong Email");
43             }
44             if (!passwordEncoder.matches(password, userApp.getPassword())) {
45                 throw new NotFoundException("Wrong Password");
46             }
47             authenticationManager.authenticate(new UsernamePasswordAuthenticationToken(username, password));
48         } catch (DisabledException e) {
49             throw new Exception("USER_DISABLED", e);
50         } catch (BadCredentialsException e) {
51             throw new Exception("INVALID_CREDENTIALS", e);
52         }
53     }
54
55     @PostMapping("/login")
56     public ResponseEntity<?> authenticate(@RequestBody @Valid AuthRequest authRequest) throws Exception {
57         authenticate(authRequest.getEmail(), authRequest.getPassword());
58         final UserDetails userDetails = authService.loadUserByUsername(authRequest.getEmail());
59         final String token = jwtService.generateToken(userDetails);
60         AuthResponse authResponse = AuthResponse.builder()
61             .token(token)
62             .userInfo(modelMapper.map(userDetails, UserResponse.class))
63             .build();
64         ApiResponse<?> response = ApiResponse.builder()
65             .message("user login successfully")
66             .payload(authResponse)
67             .status(HttpStatus.OK)
68             .statusCode(HttpStatus.OK.value())
69             .time(LocalDateTime.now())
70             .build();
71         return ResponseEntity.status(HttpStatus.OK).body(response);
72     }
73 }
```

```

1  @PostMapping("/register")
2      public ResponseEntity<?> register(@RequestBody @Valid UserRequest userRequest , @RequestParam(defaultValue = "false") Boolean withGoogle ) {
3      ApiResponse<?> response = ApiResponse.builder()
4          .message("Register successfully")
5          .payload(authService.register(userRequest,withGoogle))
6          .status(HttpStatus.OK)
7          .statusCode(HttpStatus.OK.value())
8          .time(LocalDateTime.now())
9          .build();
10     return ResponseEntity.status(HttpStatus.OK).body(response);
11 }
12
13 @PostMapping("/resendOtp")
14 public ResponseEntity<?> resendCode(String email) {
15     authService.resendCode(email);
16     ApiResponse<?> response = ApiResponse.builder()
17         .message("Code has been send")
18         .status(HttpStatus.OK)
19         .statusCode(HttpStatus.OK.value())
20         .time(LocalDateTime.now())
21         .build();
22     return ResponseEntity.status(HttpStatus.OK).body(response);
23 }
24
25 @PutMapping("forgetPassword")
26 @Operation(summary = "forget password")
27 public ResponseEntity<?> forgetPassword(@RequestParam @Valid String email, @RequestBody @Valid ForgetPasswordRequest forgetPasswordRequest) {
28     authService.forgetPassword(email, forgetPasswordRequest);
29     ApiResponse<?> response = ApiResponse.builder()
30         .message("Password was changed successful")
31         .status(HttpStatus.OK)
32         .statusCode(HttpStatus.OK.value())
33         .time(LocalDateTime.now())
34         .build();
35     return ResponseEntity.status(HttpStatus.OK).body(response);
36 }
37
38 @SecurityRequirement(name = "bearerAuth")
39 @PutMapping("/resetPassword")
40 public ResponseEntity<?> forgetPassword( @RequestBody @Valid ForgetPasswordRequest forgetPasswordRequest) {
41     authService.resetPassword(forgetPasswordRequest);
42     ApiResponse<?> response = ApiResponse.builder()
43         .message("Password was changed successful")
44         .status(HttpStatus.OK)
45         .statusCode(HttpStatus.OK.value())
46         .time(LocalDateTime.now())
47         .build();
48     return ResponseEntity.status(HttpStatus.OK).body(response);
49 }
50
51 @PutMapping("/verify")
52 public ResponseEntity<?> verifyOtpCode(String otpCode) {
53     authService.verifyOtpCode(otpCode);
54     ApiResponse<?> response = ApiResponse.builder()
55         .message("Code has been verified")
56         .status(HttpStatus.OK)
57         .statusCode(HttpStatus.OK.value())
58         .time(LocalDateTime.now())
59         .build();
60     return ResponseEntity.status(HttpStatus.OK).body(response);
61 }
62 }
63
64

```

Announcement

```
1 package org.example.classsphereapi.controller;
2
3 import io.swagger.v3.oas.annotations.Operation;
4 import io.swagger.v3.oas.annotations.security.SecurityRequirement;
5 import jakarta.validation.Valid;
6 import lombok.AllArgsConstructor;
7 import org.example.classsphereapi.exception.NotFoundException;
8 import org.example.classsphereapi.model.dto.request.AnnouncementRequest;
9 import org.example.classsphereapi.model.dto.response.ApiResponse;
10 import org.example.classsphereapi.service.AnnounceService;
11 import org.hibernate.validator.constraints.UUID;
12 import org.springframework.http.HttpStatus;
13 import org.springframework.http.ResponseEntity;
14 import org.springframework.web.bind.annotation.*;
15
16 import java.time.LocalDateTime;
17
18 @RestController
19 @AllArgsConstructor
20 @RequestMapping("/api/v1/announcements")
21 @SecurityRequirement(name = "bearerAuth")
22 public class AnnouncementController {
23     private final AnnounceService announceService;
24
25     @PostMapping
26     @Operation(summary = "Create announcement")
27     public ResponseEntity<?> createAnnounce(@Valid @RequestBody AnnouncementRequest announcementRequest) {
28         if (announcementRequest == null) throw new NotFoundException("Class id " + null + " not found");
29         ApiResponse<?> response = ApiResponse.builder()
30             .message("Announcement has been created successfully.")
31             .payload(announceService.createAnnounce(announcementRequest))
32             .status(HttpStatus.CREATED)
33             .statusCode(HttpStatus.CREATED.value())
34             .time(LocalDateTime.now())
35             .build();
36         return ResponseEntity.status(HttpStatus.CREATED).body(response);
37     }
38
39     @GetMapping("/class/{classId}")
40     @Operation(summary = "Get all announcement by class id")
41     public ResponseEntity<?> getAllAnnouncements(@PathVariable @UUID String classId) {
42         ApiResponse<?> response = ApiResponse.builder()
43             .message("Announcements retrieved successfully.")
44             .payload(announceService.getAllAnnouncement(classId))
45             .status(HttpStatus.OK)
46             .statusCode(HttpStatus.OK.value())
47             .time(LocalDateTime.now())
48             .build();
49         return ResponseEntity.status(HttpStatus.OK).body(response);
50     }
51 }
```

```

1  @PutMapping("/{announcementId}")
2  @Operation(summary = "Update announcement by id")
3  public ResponseEntity<> updateAnnounce(@PathVariable @UUID String announcementId, @Valid @RequestBody AnnouncementRequest announcementRequest) {
4      ApiResponse<> response = ApiResponse.builder()
5          .message("Announcement has been updated successfully.")
6          .payload(announceService.updateAnnounceById(announcementId, announcementRequest))
7          .status(HttpStatus.OK)
8          .statusCode(HttpStatus.OK.value())
9          .time(LocalDateTime.now())
10         .build();
11     return ResponseEntity.status(HttpStatus.OK).body(response);
12 }
13
14 @DeleteMapping("/class/{classId}/announcement/{announcementId}")
15 @Operation(summary = "Delete announcement by id")
16 public ResponseEntity<> deleteAnnouncementById(@PathVariable @UUID String classId, @PathVariable @UUID String announcementId) {
17     ApiResponse<> response = ApiResponse.builder()
18         .message("Announcement has been deleted successfully.")
19         .payload(announceService.deleteAnnouncementById(classId, announcementId))
20         .status(HttpStatus.OK)
21         .statusCode(HttpStatus.OK.value())
22         .time(LocalDateTime.now())
23         .build();
24     return ResponseEntity.status(HttpStatus.OK).body(response);
25 }
26
27 @GetMapping("/class/{classId}/announcement/{announcementId}")
28 @Operation(summary = "Get announcement by id")
29 public ResponseEntity<> getAnnouncementByIdInClass(@PathVariable @UUID String classId, @PathVariable @UUID String announcementId) {
30     ApiResponse<> response = ApiResponse.builder()
31         .message("Announcements retrieved successfully.")
32         .payload(announceService.getAnnounceByIdInClass(classId, announcementId))
33         .status(HttpStatus.OK)
34         .statusCode(HttpStatus.OK.value())
35         .time(LocalDateTime.now())
36         .build();
37     return ResponseEntity.status(HttpStatus.OK).body(response);
38 }
39
40 @GetMapping("/class/announcement/{announcementId}")
41 @Operation(summary = "Get announcement by id")
42 public ResponseEntity<> getAnnouncementById(@PathVariable @UUID String announcementId) {
43     ApiResponse<> response = ApiResponse.builder()
44         .message("Announcements retrieved successfully.")
45         .payload(announceService.getAnnounceById(announcementId.toString()))
46         .status(HttpStatus.OK)
47         .statusCode(HttpStatus.OK.value())
48         .time(LocalDateTime.now())
49         .build();
50     return ResponseEntity.status(HttpStatus.OK).body(response);
51 }
52
53 }
54

```