​

ព្រះរាជាណាចក្រកម្ពុជា

ជាតិ សាសនា ព្រះមហាក្សត្រ

3



ក្រសួងការងារ និងបណ្តុះបណ្តាលវិជ្ជាជីវៈ

វិទ្យាស្ថានពហុបច្ចេកទេសភូមិភាគតេជោសែនបាត់ដំបង

**Regional Polytechnic Institute Techo Sen Battambang**

របាយការណ៍ស្រាវជ្រាវបញ្ចប់ការសិក្សា

កម្រិត ៖​ សញ្ញាបត្រជាន់ខ្ពស់បច្ចេកទេស

ឯកទេស ៖ ព័ត៌មានវិទ្យា ជំនាន់ទី ១៧

ប្រធានបទ ៖ ប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី)

(Short Course Management System)

​​

គ្រូបច្ចេកទេសដឹកនាំ៖ លោក លាង គឹមលន

គ្រូបច្ចេកទេសជំនួយ៖ លោក ផាន់​ ធុល

**និស្សិតឈ្មោះ ​ ១. ឡេង​ គឹមហេង**

**២. សេង​​ គឹមហេង**

**៣. សាត​ ពុទ្ធារិទ្ធ**

**៤.​​ ភឿន​ ធារ៉ា**

**៥. ស៊ិន​ សក្តិនីតា**

**៦. អ៊ុយ​ ច័ន្ទគ្រឹស្នា**

ឆ្នាំ ២០២៥



រៀបរៀងដោយ ៖

* ភី ជីងពីន
* ដូណាត្រាំ

ផលិតនៅឆ្នាំ ២០១៨

Bottom: 2.5cm

Limon R 1, Size 24

គណៈកម្មការវាយតម្លៃរបាយការណ៍ស្រាវជ្រាវ

១. សាស្រ្តាចារ្យ………………………………………ហត្ថលេខា…………………………ប្រធាន

២. សាស្រ្តាចារ្យ………………………………………ហត្ថលេខា…………………………អនុប្រធាន

៣. សាស្រ្តាចារ្យ………………………………………ហត្ថលេខា…………………………សមាជិក

៤. សាស្រ្តាចារ្យ………………………………………ហត្ថលេខា…………………………សមាជិក

៥. សាស្រ្តាចារ្យ………………………………………ហត្ថលេខា…………………………សមាជិក

៦. សាស្រ្តាចារ្យ………………………………………ហត្ថលេខា…………………………សមាជិក

៧. សាស្រ្តាចារ្យ………………………………………ហត្ថលេខា…………………………សមាជិក

មតិយោបល់ប្រធានគណៈកម្មការ

សេចក្តីអំណះអំណាង

យើងខ្ញុំទាំងអស់គ្នាជានិស្សិតថ្នាក់សញ្ញាបត្រជាន់ខ្ពស់បច្ចេកទេស ឯកទេស​​ព័ត៌មានវិទ្យា ជំនាន់ទី១៧​ ក្នុងដេប៉ាតឺម៉ង់ព័ត៌មានវិទ្យា នៃវិទ្យាស្ថានពហុបច្ចេកទេសភូមិភាគតេជោសែនបាត់ដំបង សូមធ្វើការធានាអះអាងថា របាយការណ៍ស្រាវជ្រាវបញ្ចប់ការសិក្សានេះ គឺពិតជាមានភាពដើមទាំងស្រុង។ កិច្ចការដែលយើងខ្ញុំទាំងអស់គ្នាបានស្រាវជ្រាវចេញជាលទ្ធផល​​ប្រកបដោយផ្លែផ្កាគឺមិនធ្លាប់បានប្រគល់ជូនគ្រឹះស្ថានសិក្សា ឬស្ថាប័នស្រាវជ្រាវណាមួយនោះឡើយ។ យើងខ្ញុំទាំងអស់គ្នា សូមទទួលខុសត្រូវចំពោះមុខវិទ្យាស្ថាន និងច្បាប់ជាធរមាន ​ក្នុងករណី​បានពិនិត្យឃើញថា មានការក្លែងបន្លំ លួចចម្លងឯកសារ ឬស្នាដៃពីអ្នកដទៃ។

ថ្ងៃ...........................ខែ...................ឆ្នាំ​ ម្សាញ់ សប្តស័ក ព.ស.២៥៦៩

បាត់ដំបង ថ្ងៃទី..........ខែ.................ឆ្នាំ២០២៥

ហត្ថលេខា និងឈ្មោះនិស្សិត

.................................

ឡេង គឹមហេង

.................................

សេង​ គឹមហេង

...................................

សាត​ ពុទ្ធារិទ្ធ

.................................

ភឿន​ ធារ៉ា

.................................

ស៊ិន សក្តិនីតា

...................................

អ៊ុយ ច័ន្ទគ្រឹស្នា

សេចក្តីអំណះអំណាងរបស់គ្រូបច្ចេកទេសដឹកនាំ

ខ្ញុំបាទឈ្មោះ លាង គឹមលន ​ ជាគ្រូបច្ចេកទេសដឹកនាំសរសេររបាយការណ៍ស្រាវជ្រាវក្រោមប្រធានបទ **ប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី)** សូម**​​**បញ្ជាក់ថា ក្រុមនិស្សិត **ឈ្មោះ ឡេង​ គឹមហេង, សេង​ គឹមហេង , សាត ពុទ្ធារិទ្ធ , ភឿន ធារ៉ា​ , ស៊ិន​ សក្តិនីតា និង អ៊ុយ ច័ន្ទគ្រឹស្នា** ជានិស្សិតថ្នាក់សញ្ញាបត្រជាន់ខ្ពស់បច្ចេកទេស ឯកទេស​​ព័ត៌មានវិទ្យា ជំនាន់ទី១៧ ក្នុងដេប៉ាតឺម៉ង់ព័ត៌មានវិទ្យា នៃវិទ្យាស្ថានពហុបច្ចេកទេសភូមិភាគតេជោសែនបាត់ដំបង ពិតជាបានរៀបរៀង ចងក្រង និងសរសេររបាយការណ៍ស្រាវជ្រាវក្រោមប្រធានបទខាងលើ ពិតប្រាកដមែន។

ថ្ងៃ............................ខែ.................ឆ្នាំ​ ម្សាញ់ សប្តស័ក​ ព.ស.២៥៦៩

បាត់ដំបង ថ្ងៃទី..........ខែ................ឆ្នាំ២០២៥

ហត្ថលេខា និងគ្រូបច្ចេកទេសដឹកនាំ

​​ លាង​ គឹមលន

សេចក្តីថ្លែងអំណរគុណ

យើងខ្ញុំឈ្មោះ **ឡេង​ គឹមហេង, សេង​ គឹមហេង , សាត ពុទ្ធារិទ្ធ​ , ភឿន ធារ៉ា​ , ស៊ិន​ សក្តិនីតា និង អ៊ុយ ច័ន្ទគ្រឹស្នា** ជានិស្សិតថ្នាក់សញ្ញាបត្រជាន់ខ្ពស់បច្ចេកទេស ឯកទេស​​ព័ត៌មានវិទ្យា ជំនាន់ទី១៧ ក្នុងដេប៉ាតឺម៉ង់ព័ត៌មានវិទ្យា នៃវិទ្យាស្ថានពហុបច្ចេកទេសភូមិភាគតេជោសែនបាត់ដំបង សូមថ្លែងអំណរគុណយ៉ាងជ្រាលជ្រៅបំផុតចំពោះ លោកឪពុក អ្នកម្តាយ ដែលបានចិញ្ចឹមបីបាច់ថែរក្សាជួយជ្រំជ្រែង និងផ្គត់ផ្គង់ទំនុកបម្រុង អស់ពីកម្លាំងកាយចិត្ត ផ្តល់នូវសម្ភារៈគ្រប់បែបយ៉ាងក្នុងការគាំទ្រការសិក្សារបស់យើងខ្ញុំរហូតបានបញ្ចប់ជាស្ថាពរ។

ជាងនេះទៅទៀត យើងខ្ញុំទាំងអស់គ្នា សូមថ្លែងអំណរអរគុណយ៉ាងជ្រាលជ្រៅបំផុតចំពោះថ្នាក់ដឹកនាំក្រសួង​​ការងារ​ និងបណ្តុះបណ្តាលវិជ្ជាជីវៈ​ និងគណៈគ្រប់គ្រង មន្ត្រីរាជការ និងគ្រូបច្ចេកទេស​ទាំងអស់ នៃវិទ្យាស្ថានពហុបច្ចេកទេសភូមិភាគតេជោសែនបាត់ដំបង ជាពិសេស ​**លោក​ ឈឹម​ ធឿន​** នាយកវិទ្យាស្ថានដែលបានបង្កើតថ្នាលអប់រំបណ្តុះបណ្តាលបច្ចេកទេស​​ និងវិជ្ជាជីវៈ​ប្រកបដោយគុណភាព ប្រសិទ្ធភាព និងសមិទ្ធផលខ្ពស់។

យើងខ្ញុំសូមថ្លែងអំណរគុណយ៉ាងខ្ពង់ខ្ពស់ទៅដល់ វិទ្យាស្ថានពហុបច្ចេកទេសភូមិភាគតេជោសែនបាត់ដំបង​ដែលបាន ផ្តល់ទីកន្លែង សម្រាប់ការស្រាវជ្រាវ​​ព័ត៌មានទិន្នន័យ និងបទពិសោធន៍ល្អៗ ដើម្បីសរសេររបាយការណ៍ស្រាវជ្រាវបញ្ចប់ការសិក្សា រហូតសម្រេចបានលទ្ធផលជាស្ថាពរ។

ក្នុងន័យនេះដែរ យើងខ្ញុំក៏សូមថ្លែងអំណរគុណយ៉ាងក្រៃលេងចំពោះ លោក **លាង គឹមលន** និងលោក **ផាន់ ធុល** ដែលជាគ្រូបច្ចេកទេសដឹកនាំ និងគ្រូបច្ចេកទេសជំនួយការដ៏ល្អ ក្នុងការជួយណែនាំពន្យល់ និងបង្ហាត់បង្ហាញពីរបៀបសរសេររបាយការណ៍ស្រាវជ្រាវយ៉ាងអស់ពីកម្លាំងកាយចិត្ត រហូតដល់ទីបញ្ចប់។

ជាកិច្ចបញ្ចប់​​ យើងខ្ញុំទាំងអស់​គ្នា សូមតម្កល់ទុកស្នាដៃនេះជូនដល់អ្នកអាន អ្នកស្រាវជ្រាវ និងអ្នកសរសេររបាយការណ៍ស្រាវជ្រាវជំនាន់ក្រោយៗ ទទួលបាននូវសារៈប្រយោជន៍​ និងខ្លឹមសារសម្រាប់យកទៅប្រើប្រាស់តាមតម្រូវការ។ យើងខ្ញុំទាំងអស់​គ្នា រង់ចាំទទួលនូវការរិះគន់ និងស្ថាបនាពីគ្រប់មជ្ឈដ្ឋានទាំងអស់។

​​សូមគោរពជូនពរប្រសិទ្ធពរជ័យ សេរីសួស្ដី ជោគជ័យគ្រប់ភារកិច្ច និងជួបប្រទះតែសេចក្តីសុខសេចក្ដីចម្រើនព្រមទាំងពុទ្ធពរទាំង៥ប្រការ​ គឺអាយុ វណ្ណៈ សុខៈ​ ពលៈ បដិភាណៈ កុំបីឃ្លៀងឃ្លាតឡើយ។

អារម្ភកថា

ជាកិច្ចចាប់ផ្តើម យើងខ្ញុំសូមគោរពអរគុណយ៉ាងជ្រាលជ្រៅទៅដល់សំណាក់​ មិត្តសិស្សានុសិស្សទាំងអស់គ្នា ដែលបានអា​ន​ និងស្វែងយល់ពីសៀវភៅមួយក្បាលនេះ។ ដែលសៀវភៅនេះផងដែរមានចំណងជើងថា “**ប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី)**” ដែលមិត្តសិស្សានុសិស្ស កំពុងតែកាន់ជាប់នឹងដៃ គឺជាស្នាដៃមួយដែលពួកយើងខ្ញុំជាសមាជិកក្រុមទាំងអស់គ្នា បានរួមគ្នារៀបចំចងក្រងឡើងដើម្បីបញ្ចប់ការសិក្សាថ្នាក់សញ្ញាបត្រជាន់ខ្ពស់បច្ចេកទេសរបស់យើងខ្ញុំ។

លើសពីនោះទៅទៀត គឺមិនមែនមានតែគោលបំណងមួយនេះប៉ុណ្ណោះទេ ដែលជាបុព្វហេតុនាំឱ្យយើងបង្កើតជាស្នាដៃមួយនេះឡើង​។ហេតុផលមួយផ្សេងទៀតដែលក្រុមយើងខ្ញុំបានខិតខំប្រឹងប្រែងបង្កើតជាសៀវភៅមួយក្បាលនេះគឺក្នុងគោលបំណងចងក្រងជាស្នាដៃទុកជាប្រយោជន៍មួយចំពោះមិត្តអ្នកសិក្សាសម្រាប់ធ្វើការស្រាវជ្រាវដើម្បីពង្រីកចំណេះដឹងរបស់ខ្លួន ដោយទទួលយកនូវព័ត៌មានអ្វីដែលមិត្តត្រូវការពីក្នុងសៀវភៅនេះ។ ហេតុដូចនេះហើយ ទើបបានជាយើងខ្ញុំសង្ឃឹមទុកក្នុងចិត្តយ៉ាងមុតមាំថា សៀវភៅមួយក្បាលនេះ នឹងផ្តល់នូវគុណប្រយោជន៍ គុណតម្លៃជាច្រើនដល់មិត្តអ្នកសិក្សាគ្រប់រូបដែលមានបំណងចង់ក្រេបជញ្ជក់នូវចំណេះដឹងខាងផ្នែក **ប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី)** នេះ។ ពិសេសជាងនេះទៅទៀតនោះ យើងខ្ញុំក៏សង្ឃឹមផងដែរថា សៀវភៅមួយនេះ នឹងក្លាយទៅជាចំណែកដ៏សំខាន់មួយ ដើម្បីជាស្ពានជួយចម្លងមិត្តទាំងអស់គ្នាទៅកាន់ត្រើយ ជោគជ័យក្នុងជីវិតរបស់មិត្តគ្រប់រូប។

ជាការពិតណាស់ សៀវភៅមួយក្បាលនេះ គឺត្រូវបានយើងខ្ញុំរៀបចំបង្កើតឡើងបានដោយជោគជ័យទៅ តាមអ្វីដែលវិទ្យាស្ថានបានតម្រូវ ប៉ុន្តែបើទោះបីជាយ៉ាងនេះក្តី យើងខ្ញុំក៏សូមអភ័យទោសទុកជាមុននូវរាល់ កំហុសខុសឆ្គងទាំងឡាយដោយអចេតនា ដែលបានកើតឡើង មិនថាទាក់ទងទៅនឹង កំហុសឆ្គងផ្នែកអក្ខរាវិរុទ្ធ សំនួន វាចារ ឬ អត្ថន័យនៃឃ្លោងឃ្លា ល្បះ បើសិនជាមានលក្ខណៈមិនត្រឹមត្រូវ ឬមិនសមរម្យដោយប្រការណាមួយនោះ។ ទន្ទឹមនឹងការសូមអភ័យទោសនេះដែរ អ្វីដែលពួកយើងសំណូមពរពីមិត្តអ្នកសិក្សាវិញ គឺយើងខ្ញុំសូមធ្វើការទទួលស្វាគមន៍ដោយរីករាយក្រៃលេង ចំពោះរាល់ការរិះគន់ក្នុងន័យស្ថាបនាទាំងឡាយដែលទាក់ទងទៅនឹងសៀវភៅមួយក្បាលនេះ ដើម្បីរួមគ្នាឈានកាន់តែកៀកទៅនឹងភាពឥតខ្ចោះមួយ។

សរុបជារួមមក យើងខ្ញុំជាសមាជិកក្រុមទាំងអស់គ្នាសូមគោរពជូនពរដល់មិត្តសិស្សានុសិស្សគ្រប់រូបឱ្យជួបតែសេចក្តីសុខ មានសុភមង្គលគ្រប់ក្រុមគ្រួសារ ចាកផុតពីជំងឺទាំងពួង មានសុខភាពល្អ មានប្រាជ្ញាឈ្លាសវៃ បានសម្រេចនូវរាល់បំណងប្រាថ្នា ទទួលបានជោគជ័យក្នុងឆាកជីវិតជារៀងដរាបតទៅ។

មូលន័យសង្ខេប

ដោយហេតុថាបច្ចេកវិទ្យានៅក្នុងពិភពលោកយើងសព្វថ្ងៃនេះ ចេះតែដើរទៅមុខនិង​មានការរីកចម្រើនជឿនលឿនកាន់តែខ្លាំងឡើងៗ ជាពិសេសវិស័យវិទ្យាសាស្ត្រកុំព្យូទ័រ ដែលមានការពេញនិយមប្រើប្រាស់ក្នុងសកលលោក ហើយជាបច្ចេកវិទ្យាមួយដែលត្រូវតែផ្សាភ្ជាប់ទៅនឹងសង្គមបច្ចុប្បន្ន​ ដោយមិនអាច

ផ្តាច់ចេញបានឡើយ ។ កុំព្យូទ័រគឺអាចជួយសម្រួលដល់ការងារបានច្រើនយ៉ាង និងឆាប់រហ័សទាន់ចិត្តទៀតផង មិនត្រឹមតែប៉ុណ្ណោះក៏មានការប្រើប្រាស់សម្រាប់ធ្វើជាប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ (វគ្គខ្លី) ផងដែរ។ ការប្រើនូវប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ (វគ្គខ្លី) វាពិតជាបានជួយសម្រួលដល់ការងារ ឱ្យបានទាន់ពេលវេលា ដោយយោងតាមតម្រូវការនេះហើយទើបក្រុមនិស្សិត នៃវិទ្យាស្ថានពហុបច្ចេកទេសភូមិភាគតេជោសែនបាត់ដំបង ឯកទេស​​ព័ត៌មានវិទ្យា ជំនាន់ទី១៧ ដេប៉ាតឺម៉ង់ព័ត៌មានវិទ្យា បានធ្វើការស្នើសុំក្នុងការចុះធ្វើកម្មសិក្សាក្រោមប្រធានបទ **“ប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ (វគ្គខ្លី)”** នៅវិទ្យាស្ថានពហុបច្ចេកទេសភូមិភាគតេជោសែនបាត់ដំបង ដែលជាវិទ្យាស្ថានអប់រំ ផ្ដល់នូវសេវាកម្ម បង្រៀនបច្ចេកវិទ្យាវគ្គខ្លី និងជំនាញបច្ចេកទេសផ្សេងៗ ។ ទន្ទឹមនឹងការប្រើប្រាស់ នូវប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី)ដែលដំណើរការប្រចាំថ្ងៃនេះ មានការជួបប្រទះបញ្ហា មួយចំនួនដែលពុំទាន់បានធ្វើការដោះស្រាយនៅឡើយ។​ ក្រោយពីបានដឹង នូវរាល់បញ្ហាដែលបានជួបប្រទះកន្លងមកនោះ ការសិក្សាស្រាវជ្រាវនេះបានលើកជាការស្នើសុំក្នុងការបង្កើតប្រព័ន្ធមួយ ក្នុងគោលបំណងដើម្បីដោះស្រាយបញ្ហាទាំងនេះ ។ បន្ទាប់ពីបានសិក្សាទៅលើប្រព័ន្ធដែលបានស្នើសុំនេះរួចមក យើងសង្គេតឃើញថាប្រព័ន្ធនេះអាចដោះស្រាយបញ្ហាដែលជួបប្រទះមួយចំនួនដូចជា ងាយស្រួលក្នុងការគ្រប់គ្រងវគ្គសិក្សា រួមមានដូចជា ការចុះឈ្មោះសិស្ស ការគ្រប់គ្រងព័ត៌មាន ( គ្រូ ថ្នាក់រៀន សិស្ស ការបង់ថ្លៃ )។

បញ្ជីអក្សរកាត់

CLI : command-line interface

CSS : Cascading Style Sheets

DBMS : Database Management System

DFD : Data Flow Diagram

GUI : graphical user interface

HTML : Hyper Text Markup Language

ILO : International Labor Organization

IT : Information Technology

ILO : International Labor Organization

IDE                      : Integrated Development Environment

OOP                     : Object Oriented Programming

RPITSB : Regional Polytechnic Institute Techo Sen Battambang

SDLC : System Development Life Cycle

VB.Net : Visual Basic.Net

XML : Extensible Markup Language

XSLT : Extensible Stylesheet Language Transformations

បញ្ជីវិចិត្ររូប

រូបភាព

រូបភាព ទី២.១​ ​Systems Development Life Cycle ……………………………………៨

រូបភាព ទី ២.២ Context Diagram​ …………………………………………………………...១០

រូបភាព ទី ២.៣ និមិត្តសញ្ញា DFD​ ………………………………………………………...១១

រូបភាព ទី​ ៣.៤ System Database Design …………………………………………….២១

រូបភាព ទី ៣.៥ Short Course Management System ​………………………………..២២

រូបភាព ទី ៤.១.១ Login Form ………………………………………………………….៣០

រូបភាព ទី ៤.១.២ Interface Front End ………………………………………………..៣១

រូបភាព ទី ៤.១.២.១ បញ្ជីចុះឈ្មោះសិស្ស​​ ………………………………………………...៣១

រូបភាព ទី ៤.១.២.២ បញ្ជីព័ត៌មានគ្រូបង្រៀន ……………………………………………..៣២

រូបភាព ទី ៤.១.២.៣ បញ្ជីព័ត៌មានសិស្ស ………………………………………………….៣២

រូបភាព ទី ៤.១.២.៤ បញ្ជីព័ត៌មានវគ្គសិក្សា ……………………………………………….​៣៣

រូបភាព ទី ៤.១.២.៥ បញ្ជីព័ត៌មានមុខវិជ្ជា​ …………………………………………………៣៣

រូបភាព ទី ៤.១.២.៦ បញ្ជីរបាយការណ៍ ​…………………………………………………...៣៤

មាតិកា

ទំព័រ

[គណៈកម្មការវាយតម្លៃរបាយការណ៍ស្រាវជ្រាវ i](#_Toc201760487)

[សេចក្តីអំណះអំណាង ii](#_Toc201760488)

[សេចក្តីអំណះអំណាងរបស់គ្រូបច្ចេកទេសដឹកនាំ iii](#_Toc201760489)

[សេចក្តីថ្លែងអំណរគុណ iv](#_Toc201760490)

[អារម្ភកថា v](#_Toc201760491)

[មូលន័យសង្ខេប vi](#_Toc201760492)

[បញ្ជីអក្សរកាត់ vii](#_Toc201760493)

[បញ្ជីវិចិត្ររូប viii](#_Toc201760494)

[មាតិកា](#_Toc201760495) ix

[ជំពូក​​ ​១](#_Toc201760496) 1

[សេចក្តីផ្តើម 1](#_Toc201760497)

[១.១ សាវតាៈ 1](#_Toc201760498)

[១.២ មូលហេតុនៃការស្រាវជ្រាវៈ 2](#_Toc201760499)

[១.៣​​​​​​​ គោលបំណងនៃការស្រាវជ្រាវៈ 2](#_Toc201760500)

[១.៤ សារៈសំខាន់នៃការស្រាវជ្រាវ: 3](#_Toc201760501)

[១.៥ វិសាលភាព និងដែនកំណត់: 3](#_Toc201760502)

[១.៦ រចនាសម្ព័ន្ធនៃការស្រាវជ្រាវៈ 4](#_Toc201760503)

[ជំពូក​​ ២](#_Toc201760504) 5

[រំលឹកទ្រឹស្តី 5](#_Toc201760505)

[២.១ សញ្ញាណទូទៅនៃប្រព័ន្ធព័ត៌មាន 5](#_Toc201760506)

[២.១.២ អ្វីជា Visual Studio? 5](#_Toc201760507)

[២.១.២ អ្វីជា Microsoft Visual Basic? 6](#_Toc201760508)

[២.១.៣ អ្វីជា DBMS? 7](#_Toc201760509)

[២.១.៤ អ្វីជា Microsoft Access 2019 ? 7](#_Toc201760510)

[២.១.៥ អ្វីជាទិន្នន័យ(Data): 7](#_Toc201760511)

[២.១.៦អ្វីជាព័ត៌មាន (Information)? 7](#_Toc201760512)

[២.១.៧ អ្វីជាប្រព័ន្ធ (System)? 7](#_Toc201760513)

[២.១.៨ អ្វីជាប្រព័ន្ធព័ត៌មាន (Information System)? 8](#_Toc201760514)

[២.១.៩ អ្វីជាព័ត៌មានវិទ្យា (Information Technology)? 8](#_Toc201760515)

២.២​​ ដំណើរការនៃការបង្កើតប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី) (Short Course Management System ................................................................8

២.២.១ ការរៀបចំផែនការនៃប្រព័ន្ធ (System Planning) .........................9

[២.២.២ ការវិភាគប្រព័ន្ធ (System Analysis): 9](#_Toc201760516)

[២.២.៣ ការរចនាប្រព័ន្ធ (System Design): 13](#_Toc201760517)

[២.២.៤ ការអនុវត្តន៍ប្រព័ន្ធ (System Implementation): 15](#_Toc201760518)

[២.២.៥ ការប្រតិបត្តិ និងទ្រទ្រង់ប្រព័ន្ធ System Operation and Support: 15](#_Toc201760519)

[ជំពូក​​ ៣](#_Toc201760520) 16

[វីធីសាស្រ្តនៃការសា្រវជ្រាវ 16](#_Toc201760521)

[៣.១ ការរៀបចំការស្រាវជ្រាវ (Research Design) 16](#_Toc201760522)

[៣.១.១ ជំហានទី១ ពិភាក្សា និងសិក្សាទៅលើគំរោងការងារ 17](#_Toc201760523)

[៣.១.២ ជំហានទី២ ធ្វើសម្ភាសន៍ ប្រមូលទិន្នន័យ និងរៀបចំឯកសារ 17](#_Toc201760524)

[៣.១.៣ ជំហានទី៣ ការសិក្សាអំពីតម្រូវការប្រព័ន្ធថ្មី 17](#_Toc201760525)

[៣.១.៤ ជំហានទី៤ វិភាគវែកញែករកវិធីដោះស្រាយ 17](#_Toc201760526)

[៣.១.៥ ជំហានទី៥: បង្កើតប្រព័ន្ធថ្មី និង សរសេរកូដ 18](#_Toc201760527)

[៣.១.៦ ជំហានទី៦ ដំណើរការនៃប្រព័ន្ធ និងថែទាំប្រព័ន្ធ 18](#_Toc201760528)

[៣.២ វិធីសាស្រ្តស្រាវជ្រាវ (Research Methodology) 18](#_Toc201760529)

[៣.២.១ ប្រភេទទិន្នន័យ 19](#_Toc201760530)

[៣.២.២ វិធីសាស្រ្តប្រមូលទិន្នន័យ 19](#_Toc201760531)

[៣.៣ ការវិភាគទិន្នន័យ 19](#_Toc201760532)

[៣.៤ លទ្ធផលនៃការវិភាគ 20](#_Toc201760533)

[៣.៤.១ លទ្ធផលនៃការសិក្សាលើប្រព័ន្ធចាស់ 20](#_Toc201760534)

[៣.៤.២ លទ្ធផលនៃដំណោះស្រាយ 21](#_Toc201760535)

[៣.៤.៣ ការគ្រោងនៃប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី) 21](#_Toc201760536)

[៣.៤.៤ ការគ្រោងនៃ Physical Database សម្រាប់ប្រព័ន្ធថ្មី 22](#_Toc201760537)

[៣.៤.៥ លទ្ធផលនៃការរចនាប្រព័ន្ធ 22](#_Toc201760538)

[៣.៥ Data Dictionary For Table 23](#_Toc201760539)

[៣.៥.១ Table Student 23](#_Toc201760540)

[៣.៥.២ Table Register 24](#_Toc201760541)

[៣.៥.៣ Table Teacher 25](#_Toc201760542)

[៣.៥.៤ Table Subject 25](#_Toc201760543)

[៣.៥.៥ Table Schedule 26](#_Toc201760544)

[៣.៥.៦ Table Room 26](#_Toc201760545)

[៣.៥.៧ Table PaymentStatus 27](#_Toc201760546)

[៣.៥.៨ Table CourseStatus 27](#_Toc201760547)

[៣.៥.៩ Table Payment 28](#_Toc201760548)

[៣.៥.៣ Table Teacher 29](#_Toc201760549)

[ជំពូក​​ ៤](#_Toc201760550) 30

[លទ្ធផលនៃការស្រាវជ្រាវ 30](#_Toc201760551)

[៤.១ ការគ្រោងបង្កើត User Interface 30](#_Toc201760552)

[៤.១.១ Login Form 30](#_Toc201760553)

[៤.១.២ Interface Front End 30](#_Toc201760554)

[ជំពូក​​ ៥](#_Toc201769290) 35

[សន្និដ្ឋាន និង ការផ្តល់អនុសាសន៏](#_Toc201769291) 35

[៥.១ សេចក្តីសន្និដ្ឋាន](#_Toc201769292) 35

[៥.២ ការផ្តល់អនុសាសន៏](#_Toc201769293) 36

**ឯកសារយោង**

**ឧបសម្ព័ន្ធ**

ជំពូក​​ ​១

សេចក្តីផ្តើម

១.១ សាវតាៈ

នៅឆ្នាំ ១៩៩៣ ក្រោមកិច្ចសហប្រតិបត្តិការរវាងអង្គការពលកម្មអន្តរជាតិ (ILO) និងក្រសួងរ៉ែ និងថាមពល បានបង្កើតវិទ្យាស្ថានពហុបច្ចេកទេសភូមិភាគតេជោសែនបាត់ដំបងឡើងក្នុងគោលបំណងផ្តល់នូវចំណេះ ជំនាញ និងជួយរកការងារអោយដល់សិស្សានុសិស្សព្រមទាំងប្រជាជនទូទៅផងដែរ។

តាមអនុក្រឹត្យលេខ ១១៥ អនក្រ.បក ក្នុងថ្ងៃទី ១៧ ខែ មីនា ឆ្នាំ២០១៤ វិទ្យាស្ថានពហុបច្ចេកទេសខេត្តបាត់ដំបង បានកែប្រែក្លាយទៅជា វិទ្យាស្ថានពហុបច្ចេកទេសភូមិភាគតេជោសែនបាត់ដំបង(Regional Polytechnic Institute Techo Sen Battambang)។

​​អំឡុងពេលបច្ចុប្បន្ននេះ មិនថានៅអង្គការណា មជ្ឈមណ្ឌលបច្ចេកទេសណា សាលាណា ឬ ស្ថាប័នណាទេសុទ្ធតែមានសិស្សចូលមកសិក្សា ទទួលនូវការអប់រំទាំងអស់។ ដូច្នេះហើយ​ ដើម្បីឱ្យប្រតិបត្តិការរបស់អង្គភាពនោះអាចដំណើរកាទៅបានប្រកបដោយភាពជោគជ័យ ក្រៅពីបុគ្គលិកមានសមត្ថភាព​ មានវិន័យ​និងចក្ខុវិស័យច្បាស់លាស់ទៅហើយនោះ គឺត្រូវមានប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សា ប្រព័ន្ធអប់រំ ប្រព័ន្ធសេដ្ឋកិច្ច និង ការអនុវត្តន៍ទៅលើបែបបទព្រមទាំងគោលការណ៍ផ្សេងៗបន្ថែមទៀត ក្នុងការគ្រប់គ្រងសិស្សឱ្យកាន់តែល្អប្រសើរ ពិសេសការប្រើបច្ចេកវិទ្យាព័ត៌មានវិទ្យា។

ដើម្បីធានារាល់សកម្មភាពប្រព្រឹត្តិទៅ នៃការគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី)របស់វិទ្យាស្ថានពហុបច្ចេកទេសភូមិភាគតេជោសែនបាត់ដំបង ក្រុមយើងខ្ញុំបានធ្វើការបង្កើតឡើងនូវ ប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី) ដោយធ្វើការចុះ និងកត់ត្រាឈ្មោះចូលទៅក្នុងប្រព័ន្ធគ្រប់គ្រង តាមរយៈកុំព្យូទ័រ (Computer) ដែលបានប្រើប្រាស់នូវកម្មវិធី Visual Studio មកគ្រប់គ្រងទិន្នន័យរបស់វិទ្យាស្ថាន។ ប៉ុន្តែទោះបីជាយ៉ាងនេះក្ដី ក៏ការគ្រប់គ្រងនៅមានជួបបញ្ហាមួយចំនួន ដែលបង្ករឱ្យមានភាពយឺតយ៉ាវ និងការលំបាកផងដែរ ។ បច្ចុប្បន្ននេះប្រព័ន្ធគ្រប់គ្រងព័ត៌មាន ( Information Technology ) មានការរីកចម្រើនខ្លាំងពីមួយថ្ងៃទៅមួយថ្ងៃ ដោយបានបង្កើត និងចូលរួមបង្ករ​លក្ខណៈឱ្យមានភាពងាយស្រួលដល់ការគ្រប់គ្រងលើផ្នែកផ្សេងៗ យ៉ាងទូលំទូលាយលើពិភពលោក។ ហេតុដូចនេះហើយទើបអ្នកវិទ្យាសាស្ត្រខិតខំស្រាវជ្រាវបង្កើតឡើងនូវ ភាសាសម្រាប់កម្មវិធី (Program Language)ជាច្រើន ដែលគាំទ្រទៅដល់ដំណើរការគ្រប់គ្រងផ្សេងៗ និងបានជួយដល់អ្នកគ្រប់គ្រង ក្នុងការធ្វើសេចក្ដីសម្រេចចិត្តលើបញ្ហាអ្វីមួយ។ ហើយការប្រើប្រព័ន្ធនេះទៀតសោត ដើម្បីអោយការគ្រប់គ្រងមានប្រសិទ្ធិភាពខ្ពស់ និងជាកត្តាសំខាន់ដែលជួយជម្រុញឱ្យដំណើរការ គ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី) ប្រព្រឹត្តទៅបានល្អប្រសើរ និងទទួលបានជោគជ័យ ។

ដើម្បីធានាប្រសិទ្ធិភាព និងការដោះស្រាយប្រព័ន្ធគ្រប់គ្រងនេះឱ្យមានលក្ខណៈ ប្រសើរជាងនេះ ទើបក្រុមយើងខ្ញុំជានិស្សិតនៃ វិទ្យាស្ថានពហុបច្ចេកទេសភូមិភាពតេជោសែនបាត់ដំបង បានសម្រេចចិត្តជ្រើសរើសយកប្រធានបទ **“ប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ (វគ្គខ្លី) ”** ដោយរបាយការណ៍នេះគឺមានទិសដៅដើម្បីបង្វែរកិច្ចការគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី) ដែលបច្ចុប្បន្ននេះអនុវត្តន៍តាមរយៈ Microsoft Office (Excel) កំពុងមានភាពស្មុគស្មាញ និងការលំបាកជាច្រើនយ៉ាង។ ប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី)ថ្មីនេះ មានគោលបំណងជួយសម្របសម្រួល ទៅដល់បញ្ហាខាងលើដោយឆ្លងតាមការសិក្សា និងបទពិសោធន៍លើមុខវិជ្ជា ព័ត៌មានវិទ្យារបស់ក្រុមយើងខ្ញុំក្នុងរយៈពេល២ឆ្នាំកន្លងមក ក្នុងវិទ្យាស្ថានពហុបច្ចេកទេសភូមិភាគតេជោសែនបាត់ដំបង។

១.២ មូលហេតុនៃការស្រាវជ្រាវៈ

ដោយផ្អែកលើការចុះសាកសួរនៅក្នុងវិទ្យាស្ថានផ្ទាល់ របស់ក្រុមយើងខ្ញុំបានអោយដឹងថា បច្ចុប្បន្ននេះវិទ្យាស្ថានពហុបច្ចេកទេសភូមិភាគតេជោសែនបាត់ដំបង បានកត់ត្រានូវរាល់ព័ត៌មានសិស្ស ដោយប្រើលើកម្មវិធី Microsoft Excel សម្រាប់គ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី)។ ការធ្វើបែបនេះ បានធ្វើឱ្យជួបប្រទះនូវផលលំបាកជាច្រើនក្នុងការគ្រប់គ្រងប្រព័ន្ធមួយនេះដូចជា:

* កំណើននៃទិន្នន័យច្រើនខ្លាំងពេក​ ពិបាកនឹងគ្រប់គ្រងឱ្យបានល្អត្រឹមត្រូវ។
* មានផលលំបាក​ ក្នុងការស្វែងរកទិន្នន័យ ដោយធ្វើការ Scroll ដើម្បីរកមើល។
* បើចង់ធ្វើការកែប្រែទិន្នន័យដូចជា Update, Delete, Insert ត្រូវធ្វើដោយផ្ទាល់នៅលើ​ Cell ​នីមួយៗនៃ​ Work Sheet ផ្សេងគ្នាដែលមានទំនាក់ទំនងជាមួយគ្នា។
* បណ្ដុំនៃរបាយទិន្នន័យដែលបានផ្ទុកនៅលើ Work sheet មានលក្ខណៈស្មុគស្មាញ។
* ទិន្នន័យអាចមានការភាន់ច្រឡំគ្នា ឬស្ទួនគ្នាជាយថាហេតុ។
* ទិន្នន័យដែលមាននៅក្នុងប្រព័ន្ធពុំមានសុវិត្ថិភាព ព្រោះគ្មានការកំណត់កម្មសិទ្ធច្បាស់លាស់ក្នុងការចូលប្រើប្រាស់។

១.៣​​​​​​​ គោលបំណងនៃការស្រាវជ្រាវៈ

​ ដោយហេតុតែចង់បំពេញនូវតម្រូវការរបស់វិទ្យាស្ថានពហុបច្ចេកទេសភូមិភាគតេជោសែនបាត់ដំបង​ និងទទួលបាន ការគាំទ្រពីលោកគ្រូ លាង គឹមលន ដែលជាគ្រូដឹកនាំ បានអនុញ្ញាតិឱ្យក្រុមយើងខ្ញុំបានចុះធ្វើកម្មសិក្សាស្រាវជ្រាវដើម្បីរៀបចំនូវប្រព័ន្ធថ្មី ដែលគ្រប់គ្រងបែប Application Software ដែលប្រើប្រាស់កម្មវិធីVisual Basic.net​​ (Interface) និងMicrosoft Access (Data Store) ជាប្រយោជន៍ដល់វិទ្យាស្ថាន និងបានផ្ដល់ដល់ក្រុមយើងខ្ញុំវិញនូវបទពិសោធន៍ក្នុងការសិក្សា បែបបទនៃការធ្វើការងារ ដើម្បីយកមកសរសេរជារបាយការណ៍សម្រាប់ការពារ ប្រឡងបញ្ចប់ឆ្នាំសិក្សានាពេលខាងមុខនេះ។

​យោងទៅតាមបញ្ហាទាំងឡាយខាងលើនេះ ក្រុមយើងខ្ញុំដែលជាអ្នកសិក្សាស្រាវជ្រាវបានខិតខំរកគ្រប់មធ្យោបាយដើម្បីជួយដោះស្រាយនូវបញ្ហាខាងលើ​ និងមានគោលបំណងកែប្រែប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី)ឱ្យមានលក្ខណៈកាន់តែល្អប្រសើរឡើង។

* គ្រប់គ្រងការចុះឈ្មោះសិស្សចូលរៀន
* គ្រប់គ្រងវគ្គសិក្សា
* គ្រប់គ្រងថ្នាក់រៀន
* គ្រប់គ្រងគ្រូបង្រៀន​
* គ្រប់គ្រងសិស្ស
* គ្រប់គ្រងការបង់ថ្លៃសិស្ស

ហេតុនេះហើយ ដើម្បីដោះស្រាយនូវរាល់បញ្ហាខាងលើនេះ និងស្វែងរកគន្លឹះថ្មីៗដើម្បីឱ្យការអនុវត្តន៍បានល្អប្រសើរ គឺត្រូវធ្វើការយ៉ាងយកចិត្តទុកដាក់ និងមានការពិគ្រោះយ៉ាងល្អិតល្អន់​បំផុតពីសំណាក់លោកសាស្រ្តាចារ្យដឹកនាំការសិក្សា។

១.៤ សារៈសំខាន់នៃការស្រាវជ្រាវ:

បន្ទាប់ពីបានសិក្សាស្វែងយល់ពីប្រព័ន្ធចាស់រួចមក យើងខ្ញុំបានឃើញពីផលវិបាកនៃការប្រើប្រព័ន្ធចាស់ខាងលើ ហេតុនេះហើយក្រុមយើងខ្ញុំបានបង្កើតនូវប្រព័ន្ធថ្មីមួយដោយធ្វើការកែប្រែ និងបញ្ចូលទាំងកម្មវិធីថ្មីៗដែលធ្វើឱ្យប្រព័ន្ធថ្មីនេះមានលទ្ធភាពបំពេញកិច្ចបានល្អជាងមុន ជាពិសេសនោះគឺ មានល្បឿនលឿនចំណេញពេលវេលា ។​ ក្រោយពីបង្កើតប្រព័ន្ធថ្មីនេះរួចក្រុមយើងខ្ញុំសង្ឃឹមយ៉ាងមុតមាំថា កម្មវិធីនេះប្រាកដជាមានគុណប្រយោជន៍ជាច្រើនសម្រាប់បំពេញនូវរាល់កង្វះខាត ដែលផ្នែកគ្រប់គ្រងសិស្សធ្លាប់ជួបប្រទះពីមុនមក។ ដែលប្រព័ន្ធថ្មី នេះមាន​គុណប្រយោជន៍ដូចតទៅ:

-មានការ ងាយស្រួលក្នុងការស្វែងរកនូវទិន្នន័យដែលយើងចង់បាន

- ងាយស្រួលក្នុងការបញ្ចូលទិន្នន័យរបស់សិស្សថ្មី

-អាចស្វែងរកចំនួនសិស្ស និងអាចលុបទិន្នន័យចេញបាន

- មានសុវត្ថិភាពខ្ពស់ និងមានដំណើរការលឿន

-ងាយស្រួលក្នុងការ កែប្រែ ទិន្នន័យ មិនធ្វើឱ្យបាត់បង់ទិន្នន័យ

-អាចដឹងពីចំនួនសិស្សដែលបានចូលរៀន

-ទិន្នន័យអាចរក្សាទុកបានយូរងាយស្រួលក្នុងការប្រែឬលុប

-មានភាពងាយស្រួលដល់អ្នកប្រើប្រាស់ ក្នុងការបញ្ចូលបង្ហាញទិន្នន័យ​ និងបោះពុម្ព

-កាត់បន្ថយពេលវេលាក្នុងការរុករកឯកសារ

-ព័ត៌មានទទួលបានទាន់ពេលវេលា និងមានសុពលភាព

-បំពេញតម្រូវការយ៉ាងឆាប់រហ័ស

-អាចគ្រប់គ្រង ទិន្នន័យបានច្រើនដោយមិនមានបង្ករអោយមានភាពខុសឆ្គង

-លែងមានបញ្ហាទិន្នន័យមានការភាន់ច្រឡំគ្នា ឬស្ទួនគ្នា

តាម​គុណប្រយោជន៍ខាងលើនេះបានអោយយើងឃើញថាប្រព័ន្ធនេះពិតជាបាន ផ្ដល់នូវលក្ខណៈពិសេសជាច្រើនទៅដល់អ្នកប្រើប្រាស់ទាំងអស់គ្នា។

១.៥ វិសាលភាព និងដែនកំណត់:

ជាការពិតណាស់បើយើងក្រឡែកទៅមើល ប្រព័ន្ធគ្រប់គ្រងនៅក្នុង RPITSB ទាំងមូលវាមានលក្ខណទ្រង់ទ្រាយធំ ហេតុនេះក្រុមយើងខ្ញុំមិនអាចធ្វើការគ្រប់គ្រងបាននូវប្រព័ន្ធនេះទាំងមូលបានទេ ដោយសារតែរយ:ពេលនៃការសិក្សារបស់យើងខ្ញុំមានរយ:ពេលខើចខ្លី និងសមត្ថភាពក្រុមយើងខ្ញុំនៅមានកម្រិតនៅឡើយ ដូច្នេះហើយទើបក្រុមនិស្សិតយើងខ្ញុំជ្រើសយកតែ ផ្នែកខ្លះៗនៃប្រព័ន្ធ មកបង្ហាញ ។

ព្រំដែនកំណត់នៃការសិក្សាប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី) ដែលមានសមត្ថភាពធ្វើការគ្រប់គ្រងដូចជា :

* កត់ត្រារាល់ព័ត៌មានសិស្សចូលរៀន
* គ្រប់គ្រងបញ្ជីព័ត៌មានសិស្ស
* គ្រប់គ្រងបញ្ជីព័ត៌មានគ្រូ
* គ្រប់គ្រងបញ្ជីមុខវិជ្ជា
* គ្រប់គ្រងបន្ទប់សិក្សា
* គ្រប់គ្រងម៉ោងសិក្សា

ម្យ៉ាងវិញទៀតនៅក្នុងដំណើរការប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី**)** គឺបានប្រើប្រាស់នូវ Visual Basic.Net ថែមទាំង Microsoft Office (Access) សម្រាប់ធ្វើការផ្ទុកទិន្ន័យ។

១.៦ រចនាសម្ព័ន្ធនៃការស្រាវជ្រាវៈ

នេះគឺជាប្លង់នៃការស្រាវជ្រាវ គូសវាសគំរោងនៃការស្រាវជ្រាវដើម្បីអោយអ្នកអាន យល់អំពីដំណើរការនៃការស្រាវជ្រាវទាំងមូល។

ជំពួកទី១: សេចក្តីផ្តើម

ជំពួកទី២: រំលឹកទ្រឹស្តី

ជំពួកទី៣: វិធីសាស្រ្តស្រាវជ្រាវ

ជំពួកទី៤: លទ្ធផលនៃការស្រាវជ្រាវ

ជំពួកទី៥: សន្និដ្ឋាន និង ការផ្តល់អនុសាសន៍

**​**

ជំពូក​​ ២

រំលឹកទ្រឹស្តី

២.១ សញ្ញាណទូទៅនៃប្រព័ន្ធព័ត៌មាន

២.១.១ អ្វីជា Visual Studio?

**Visual Studio** គឺ integrated development environment (IDE) ពី Microsoft ។ វា​ត្រូវ​បាន ​ប្រើ​ដើម្បី​បង្កើត​កម្មវិធី​កុំព្យូទ័រ(Computer programs) ​រួម​ទាំង គេហទំព័រ(Websites) កម្មវិធី​បណ្ដាញ(Web apps), សេវាបណ្ដាញ(Web services )​ និង ​កម្មវិធី​ទូរសព្ទ(Mobile apps ។ Visual Studio ប្រើប្រាស់Microsoft software development platforms ដូចជា Windows API , Windows Forms , Windows Presentation Foundation , Windows Store និង Microsoft Silverlight ។ វាបង្កើតទាំង កូដដើម(Native code) និង កូដគ្រប់គ្រង(Managed code) ។ Visual Studio រួមបញ្ចូល កម្មវិធីកែប្រែកូដ(Code editor) ដែលគាំទ្រ IntelliSense (សមាសធាតុបំពេញកូដ The [code completion](https://en.wikipedia.org/wiki/Code_completion) component ) ក៏ដូចជា ការកែកូដឡើងវិញ(Code refactoring) ។ ឧបករណ៍បំបាត់កំហុសរួមបញ្ចូលគ្នាធ្វើការជាកម្មវិធីបំបាត់កំហុសកម្រិតប្រភព និងជាឧបករណ៍បំបាត់កំហុសកម្រិតម៉ាស៊ីន។ ឧបករណ៍ដែលភ្ជាប់មកជាមួយផ្សេងទៀតរួមមាន code profiler អ្នករចនាសម្រាប់បង្កើត កម្មវិធី GUI អ្នករចនាគេហទំព័រ និង អ្នករចនា គ្រោងការណ៍មូលដ្ឋានទិន្នន័យ ។ វាទទួលយកកម្មវិធីជំនួយដែលពង្រីកមុខងារស្ទើរតែគ្រប់កម្រិត រួមទាំងការបន្ថែមការ គាំទ្រសម្រាប់ប្រព័ន្ធ គ្រប់គ្រងប្រភព (ដូចជា Subversion និង Git) និងការបន្ថែមឧបករណ៍ថ្មីដូចជាកម្មវិធី និពន្ធ និងអ្នករចនារូបភាពសម្រាប់ ភាសាជាក់លាក់នៃដែន ឬឧបករណ៍សម្រាប់ទិដ្ឋភាពផ្សេងទៀតនៃ វដ្តនៃការអភិវឌ្ឍន៍កម្មវិធី (ដូចជា អតិថិជន Azure DevOps ៖ Team Explorer)។Visual Studio គាំទ្រភាសាសរសេរកម្មវិធី ចំនួន 36 ផ្សេងគ្នា និងអនុញ្ញាតឱ្យកម្មវិធីសរសេរកូដ(Code editor) និងកម្មវិធីបំបាត់កំហុសដើម្បីគាំទ្រ(debugger to support) ( (ដល់កម្រិតផ្សេងៗគ្នា) ស្ទើរតែគ្រប់ភាសាសរសេរកម្មវិធី ដែលផ្តល់ សេវាកម្មជាក់លាក់ភាសាមាន។ ភាសាដែលភ្ជាប់មកជាមួយរួមមាន C , C++ , C++/CLI , Visual Basic .NET , C# , F# , JavaScript , TypeScript , XML , XSLT , HTML , និង CSS ។ គាំទ្រភាសាផ្សេងទៀតដូចជា Python , Ruby , Node.js , និង M ក្នុងចំណោមកម្មវិធីផ្សេងទៀតគឺអាចរកបានតាមរយៈ កម្មវិធីជំនួយ ។ Java និង J# ត្រូវបានគាំទ្រកាលពីអតីតកាល។មូលដ្ឋានច្រើនបំផុតនៃរៀបចំ Visual Studio ដែលជារៀបចំ សហគមន៍គឺអាចរកបានដោយឥតគិតថ្លៃ។ ។ គិតត្រឹមថ្ងៃទី 10 ខែមករា ឆ្នាំ 2023 Visual Studio 2022 គឺជាកំណែ(Update)ដែលត្រៀមរួចជាស្រេចក្នុងការផលិតនាពេល បច្ចុប្បន្ន។ ស្ទូឌីយោ Visual 2013, 2015 និង 2017 ស្ថិតនៅលើផ្នែកជំនួយបន្ថែម ខណៈពេលដែលឆ្នាំ 2019 គឺស្ថិតនៅលើ Mainstream Support ។

ការអភិវឌ្ឍនបន្តបន្ទាប់នៃកម្មវិធី Visual Studio មានដូចជា៖

* Visual Studio97 ឈ្មោះកូដ (Code name) Boston ចេញថ្ងៃទី19 ខែមីនា ឆ្នាំ1997 Version 5.0 Update Version ចុងក្រោយ Service Pack 3 ថ្ងៃ Update ចុងក្រោយថ្ងៃទី04 ខែធ្នូ ឆ្នាំ1197
* Visual Studio6.0 ឈ្មោះកូដ (Code name) Aspen ចេញថ្ងៃទី02 ខែកញ្ញា ឆ្នាំ1998 Version 6.0 Update Version ចុងក្រោយ Service Pack 6 ថ្ងៃ Update ចុងក្រោយថ្ងៃទី29 ខែមីនា ឆ្នាំ2004
* Visual Studio.Net (2002) ឈ្មោះកូដ (Code name) Rainier ចេញថ្ងៃទី13 ខែកុម្ភៈ ឆ្នាំ2002 Version 7.0 Update Version ចុងក្រោយ Service Pack 1 ថ្ងៃ Update ចុងក្រោយថ្ងៃទី08 ខែមីនា ឆ្នាំ2005
* Visual Studio.Net 2003 ឈ្មោះកូដ (Code name) Everett ចេញថ្ងៃទី24 ខែមេសា ឆ្នាំ2003 Version 7.1 Update Version ចុងក្រោយ Service Pack 1 ថ្ងៃ Update ចុងក្រោយថ្ងៃទី15 ខែសីហា ឆ្នាំ2006
* Visual Studio 2005 ឈ្មោះកូដ (Code name) Whidbey ចេញថ្ងៃទី07 ខែវិច្ឆិកា ឆ្នាំ2005 Version 8.0 Update Version ចុងក្រោយ Service Pack 1 ថ្ងៃ Update ចុងក្រោយថ្ងៃទី11 ខែ ធ្នូ ឆ្នាំ2006
* Visual Studio 2008 ឈ្មោះកូដ (Code name) *Orcas* ចេញថ្ងៃទី10 ខែវិច្ឆិកា ឆ្នាំ2007 Version 9.0 Update Version ចុងក្រោយ Service Pack 1 ថ្ងៃ Update ចុងក្រោយថ្ងៃទី11 ខែ សីហា ឆ្នាំ2008
* Visual Studio 2010 ឈ្មោះកូដ (Code name) Dev10ចេញថ្ងៃទី12 ខែមេសា ឆ្នាំ2010 Version 10.0 Update Version ចុងក្រោយ Service Pack 1 ថ្ងៃ Update ចុងក្រោយថ្ងៃទី10 ខែ មីនា ឆ្នាំ2011
* Visual Studio 2012 ឈ្មោះកូដ (Code name) *Dev11* ចេញថ្ងៃទី12 ខែកញ្ញា​ ឆ្នាំ2012 Version 11.0 Update Version ចុងក្រោយ Service Pack 1 ថ្ងៃ Update ចុងក្រោយថ្ងៃទី24 ខែសីហា ឆ្នាំ2015
* Visual Studio 2013 ឈ្មោះកូដ (Code name) Dev12ចេញថ្ងៃទី17 ខែតុលា​ ឆ្នាំ2013Version 12.0 Update Version ចុងក្រោយ Update 5 ថ្ងៃ Update ចុងក្រោយថ្ងៃទី20 ខែកក្កដា ឆ្នាំ2015
* Visual Studio 2015 ឈ្មោះកូដ (Code name) Dev14ចេញថ្ងៃទី20 ខែកក្កដា ​ឆ្នាំ2015 Version 14.0 Update Version ចុងក្រោយ Update 3 ថ្ងៃ Update ចុងក្រោយថ្ងៃទី27 ខែឧសភា ឆ្នាំ2016
* Visual Studio 2017 ឈ្មោះកូដ (Code name) Dev15 ចេញថ្ងៃទី07 ខែមីនា ឆ្នាំ2017 Version 15.0 Update Version ចុងក្រោយ 15.9.55 ថ្ងៃ Update ចុងក្រោយថ្ងៃទី13 ខែឧសភា ឆ្នាំ2023
* Visual Studio 2019 ឈ្មោះកូដ (Code name) Dev16 ចេញថ្ងៃទី02 ខែមេសា ​ ឆ្នាំ2019 Version 16.0 Update Version ចុងក្រោយ 16.11.27 ថ្ងៃ Update ចុងក្រោយថ្ងៃទី13 ខែឧសភា ឆ្នាំ2023
* Visual Studio 2019 ឈ្មោះកូដ (Code name) Dev17 ចេញថ្ងៃទី08 ខែវិច្និកា ​ឆ្នាំ2021 Version 17.0 Update Version ចុងក្រោយ 17.6.5 ថ្ងៃ Update ចុងក្រោយថ្ងៃទី11 ខែ កក្កដា ឆ្នាំ2023

២.១.២ អ្វីជា Microsoft Visual Basic?

**Microsoft Visual Basic** គឺជាការអនុវត្តរបស់ Microsoft លើ ភាសា [VB.NET](https://en-m-wikipedia-org.translate.goog/wiki/VB.NET?_x_tr_sl=en&_x_tr_tl=km&_x_tr_hl=km&_x_tr_pto=sc) និងឧបករណ៍ពាក់ព័ន្ធ និងសេវាកម្មភាសា។ វាត្រូវបានណែនាំជាមួយ Visual Studio .NET (2002) ។ ក្រុមហ៊ុន Microsoft បានដាក់ទីតាំង Visual Basic សម្រាប់ [ការអភិវឌ្ឍន៍កម្មវិធីរហ័ស](https://en-m-wikipedia-org.translate.goog/wiki/Rapid_Application_Development?_x_tr_sl=en&_x_tr_tl=km&_x_tr_hl=km&_x_tr_pto=sc)(Rapid Application Development) ។  Visual Basic អាចត្រូវបានប្រើដើម្បីសរសេរកម្មវិធីទាំងពីរ គឺកម្មវិធី Console ក៏ដូចជាកម្មវិធី GUI ផងដែរ។ ដូច Visual C# ដែរ Visual Basic ក៏គាំទ្រ Visual Studio Class Designer, From Designer និងData Designer ក្នុងចំណោមដទៃទៀត។ ក៏ដូច C# ដែរ VB.NET compiler ក៏មានជាផ្នែកនៃ .NET Framework ដែរ ប៉ុន្តែសេវាកម្មភាសាដែលអនុញ្ញាតឱ្យគម្រោង VB.NET ត្រូវបានបង្កើតឡើងជាមួយ Visual Studio គឺមានជាផ្នែកនៃកម្មវិធីក្រោយៗទៀត។

២.១.៣ អ្វីជា DBMS?

**(**Database Management System) គឺជា Software ពិសេសដើម្បីបង្កើត និងថែទាំ database និងអាចមានសមត្ថភាពបែងចែក Business application នីមួយៗដើម្បីទាញយក data (to extract data) ដែលត្រូវប្រើដោយមិនបាច់បង្កើត file ដាច់ដោយឡែក ផ្តល់និយមន័យថ្មីក្នុង Program របស់ខ្លួនក្នុងកុំព្យូទ័រមានន័យថា DBMS ជា Software ដែលប្រើសម្រាប់គ្រប់គ្រង Database ។ DBMS ដើរតួជា Interface រវាង application program និង physical data file ។ នៅពេល application program ត្រូវការ data, DBMS ស្វែងរកក្នុង database និងតំណាងឱ្យ data នោះទៅកាន់ application program នោះ។ ការប្រើ data fileធម្មតា,​ ​អ្នកកម្មវិធីត្រូវតែកំណត់ទំហំ (size) និងទម្រង់​ (format) នៃធាតុdata​នីមួយៗត្រូវបានប្រើក្នុងកម្មវិធី ហើយបន្ទាប់មកបញ្ជាកុំព្យូទ័រថាទីតាំងនៃ data fileទាំងនេះនៅកន្លែងណា។ DBMS ធ្វើឱ្យអ្នកប្រើប្រាស់ធម្មតាងាយស្រួលយល់ដឹងថាតើការងារណា នៅកន្លែងណាព្រមទាំងយល់ពីរបៀបទាញយក data ដែល ត្រូវបានផ្ទុកដោយការបែងចែក Logical និង Physical view នៃ data។ Logical view តំណាងឱ្យ data ដូចដែលបង្ហាញទៅកាន់ application របស់អ្នកកម្មវិធី ឬ application របស់អ្នកប្រើប្រាស់ធម្មតា។ ចំណែកឯ Physical view តំណាងឱ្យ data ដែលបានរៀបចំពិតប្រាកដលើ Physical storage media។ ទោះបីមាន physical view នៃ data តែមួយប៉ុណ្ណោះក៏អាចមាន Logical view ផ្សេងៗជាច្រើន ។ (ឯកសារនេះដកស្រង់ចេញពីសៀវភៅ Management Information System 7th Edition និពន្ធដោយ Kenneth C. Laudon, Jane P. Laudon ទំព័រទី២០៩) ។

២.១.៤ អ្វីជា Microsoft Access 2019 ?

កម្មវិធី Microsoft Access 2019 គឺជាកម្មវិធីមួយដែល ត្រូវបានប្រើប្រាស់ដើម្បីបង្កើតជា Database ដើម្បីរក្សាទុកទិន្នន័យទាំងឡាយ (Access 2003-FLM, 2008) ។ កម្មវិធីនេះមាន Tool គ្រប់គ្រាន់ និងត្រូវបានបង្កើតឡើងតាំងពីឆ្នាំ១៩៩២មកម្ល៉េះ។ ទោះបីជាវាមិនទូលំទូលាយដូច Database Platforms ដទៃទៀតក៏ដោយ ក៏លទ្ធភាពប្រើប្រាស់របស់វានៅតែល្អគ្រប់គ្រាន់សម្រាប់រៀបចំ data sets ខ្នាតតូចបានផងដែរ ហើយវាក៏ជា ជម្រើសដ៏ល្អសម្រាប់ការប្រើប្រាស់ផ្ទាល់ខ្លួន ។

២.១.៥ អ្វីជាទិន្នន័យ(Data):

គឺជាការពិតនៃអ្វីមួយឬក៏ជាគោលគំនិតដែលអាចជាមនុស្ស,ទីកន្លែង, ព្រឹត្តិការណ៍, សកម្មភាពមួយផ្នែកនៃវត្ថុមួយចំនួន។ Data ជាអ្វីដែលបាន Input ទៅក្នុង System ជាទិន្ន័យដើម​ ប្រើសម្រាប់ Process ជាការកត់ត្រានៃសភាព ឬលក្ខណអ្វីមួយ។

២.១.៦អ្វីជាព័ត៌មាន (Information)?

គឺជាទិន្នន័យដែលត្រូវបានគេរៀបចំធ្វើអោយប្រសើរ ឡើងវិញ និងធ្វើការបង្ហាញក្នុងទម្រង់មួយដែលនឹងត្រូវបានគេប្រើដើម្បីផ្ដល់ភាពងាយស្រួលក្នុងការធ្វើការសម្រេចចិត្ត ឬក៏សកម្មភាពអ្វីផ្សេងទៀត។ Information ជាអ្វីដែលបាន Outputជាព័ត៌មានដែលបានចេញពី Data សម្រាប់ប្រើ និងវិភាគ ។

២.១.៧ អ្វីជាប្រព័ន្ធ (System)?

គឺជារចនាសម្ព័ន្ធមួយដែលផ្សំឡើងពីធាតុផ្សេងៗដែលទាក់ទងគ្នានិងបំពេញនូវមុខងារជាក់លាក់

មួយ។

២.១.៨ អ្វីជាប្រព័ន្ធព័ត៌មាន (Information System)?

គឺជាប្រព័ន្ធមួយដែលមានមុខងារក្នុងការ ផ្តល់ព័ត៌មានសជួយម្រាប់ក្នុងការប្រតិបត្តិការណ៍ និងការគ្រប់គ្រងអង្គភាពពាណិជ្ជកម្ម ។

២.១.៩ អ្វីជាព័ត៌មានវិទ្យា (Information Technology)?

ព័តម៌ានវិទ្យា (Information Technology) គឺជាបណ្តុំនៃបច្ចេកវិទ្យាកុំព្យូទ័ររួមមានផ្នែក Hardware, Software, Telecommunication system, Data និង Information ដែលជួយឱ្យពាណិជ្ជកម្មមួយបង្កើតសេវាកម្មល្អ និងជួយដល់អ្នកគ្រប់គ្រងក្នុងការធ្វើការសម្រេចចិត្ត។

២.២​​ ដំណើរការនៃការបង្កើតប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី)

**(Short Course Management System)**

ដើម្បីជាជំហានឈានទៅរកការអភិវឌ្ឍន៍ប្រព័ន្ធគ្រប់គ្រងនេះ អោយបានសម្រេចដោយជោគជ័យយើងត្រូវអនុវត្តន៍ទៅតាមប្រព័ន្ធគម្រោងដែលយើងបានសិក្សារួច និងផ្អែកលើទ្រឹស្ដីដែលមាន ។ SDLC គឺជាដំណើរការដែលវិភាគទៅលើប្រព័ន្ធដែលមាន Software Engineers, Progr​ammers និងអ្នកប្រើប្រាស់ (End-Users) បង្កើតបានជា Information System និង Computer Application ។

System Development Life Cycle (SDLC) ចែកចេញជា 5 ដំណាក់កាលគឺ:

* ការរៀបចំផែនការនៃប្រព័ន្ធ (System Planning)
* ការវិភាគប្រព័ន្ធ(System Analysis)
* ការគ្រប់គ្រងប្រព័ន្ធ(System Design)
* ការអនុវត្តន៍ប្រព័ន្ធ (System Implementation)
* ការប្រតបត្តិ និង ទ្រទ្រង់ប្រព័ន្ធ(System Operation and Support)

Planning

Obsolete solution

Problem to be soloved

Operation and Support

Analysis

Implemented

solution

Implementation

error

to be fixed

Implementation

Acceptable

solution

Design

Problem analysis

and

solution requirements

Related problem to be sloved

New solution

to same problem

រូបភាពទី ២.១ Systems Development Life Cycle

២.២.១ ការរៀបចំផែនការនៃប្រព័ន្ធ (System Planning)

System Planning គឺជាសហការគ្នារវាងអ្នកប្រើប្រាស់​ និង​ IT Manager ដើម្បីរៀបចំផែនការយ៉ាងណាទើបបានជា​ ប្រព័ន្ធព័ត៌មានដែលអាចបម្រើសេចក្តីត្រូវការផ្នែកព័ត៌មានអោយបានរហ័សនិងមានប្រយោជន៍។​ ផ្នែកនេះគឺស្វែងរកគោលដៅ​ Technologies និង​ Application

ទាំងនោះត្រូវបានគេដាក់ជាអត្ថិភាពចម្បងហើយមានសារៈសំខាន់ចំពោះ Business ក្នុងនោះរួមបញ្ចូល ទាំងការរៀបចំប្រព័ន្ធគម្រោងប្រព័ន្ធព័ត៌មាន​ និងប្រភពព័ត៌មាន។​​​ ការរៀបចំផែនការនៃប្រព័ន្ធជាជំហាន​​ ដំបូងនៃការបង្កើតប្រព័ន្ធព័ត៌មាន​ ។ គោលបំណងសំខាន់ក្នុងផ្នែកនេះគឺ ជាការសិក្សាអំពីយុទ្ធសាស្រ្ត នៃក្រុមហ៊ុនរួចជ្រើសរើសយកប្រព័ន្ធព័ត៌មានដែលអាចយកមកប្រើ​ដើម្បី ជួយដល់ក្រុមហ៊ុន ក្នុងការអនុវត្តន៍នូវយុទ្ធសាស្រ្តរបស់ខ្លួន។​ គោលបំណងនៃការរៀបចំគម្រោងគឺជាការបញ្ជាក់ប្រភេទយ៉ាងច្បាស់លាស់ពីព្រំដែននៃ Business ឬក៏បញ្ហាតាមរយៈការអនុវត្តន៍ ការចាប់ផ្ដើមវិនិយោគទន់ដែលគេហៅ​ ថាភាពជឿជាក់បាន នៃការសិក្សា។ ការចាប់ផ្តើមវិនិយោគទន់ដំបូងគឺជាជំហាននៃការកែរកចាប់កំហុស ព្រោះបញ្ហាផ្សេង​ៗ​ នឹងប៉ះពាល់ដល់ការបញ្ចូលពីដំណើរការនៃការអនុវត្តន៍របស់យើង។​ អ្វីដែលផលិតបាន ឬដែលបានចែកចាយហើយគឺជារបាយការណ៍ដែលពិពណ៌នាអំពី Business ការរំលឹកឡើងវិញ​ រវាងប្រាក់ចំណេញ​ និងតម្លៃដែលយើង​សង្ឃឹមទុកជាមុន និងការផ្ដល់យោបល់វាយតម្លៃ នៃសកម្មភាព ពឹងផ្អែកទៅលើសេដ្ឋកិច្ចបច្ចេកទេស និងកត្តាជាក់ស្ដែង។

​​​ រីឯអ្នកវិភាគប្រព័ន្ធ ( System Analyst ) និងទទួលបាននូវសំណើរស្ដីអំពីការផ្លាស់ប្ដូរឬក៏ពង្រីកនូវប្រព័ន្ធ។ ជំហ៊ានដំបូងគឺការកើតឡើងដោយគិតពិចារណានិងសិក្សាបន្ថែម អ្នកវិភាគប្រព័ន្ធ ( System Analyst ) អាចស្វែងយល់ផងដែរនូវ System Functions ដែលល្អ រីឯអ្នកប្រើប្រាស់ ( Users ) វិញគេត្រូវធ្វើការបង្ហាត់បង្រៀនច្រើន។

នៅក្នុងស្ថានភាពខ្លះអ្នកប្រហែលជារួមបញ្ចូលផងដែរពីការរំលឹក នូវទំហំនៃប្រព័ន្ធពេញលេញគឺវាមានសារៈសំខាន់ណាស់។ ប្រសិនបើដំណើរការអនុវត្តន៍នៅតែបន្តជំហ៊ានបន្តបន្ទាប់គឺការវិភាគប្រព័ន្ធ​ ( System Analyst )។

(ឯកសារនេះដកស្រង់ចេញពីសៀវភៅ System Analyst and Design 4th Edition និងនិពន្ធដោយ​ Gray B. Shelly, Thomas J. Rosenblatt ទំព័រទី ១.២០)

២.២.២ ការវិភាគប្រព័ន្ធ (System Analysis):

System Analysis គឺវិភាគសិក្សាលំអិតបញ្ហាការលំបាកដើម្បីកំណត់តម្រូវការចាំបាច់ក្នុងការបំបាក់បញ្ហា និងតម្រូវការសម្រាប់ការកសាងប្រព័ន្ធព័ត៌មាន។ គោលបំណងចាំបាច់ក្នុងដំណាក់កាលនេះគឺការវិភាគយ៉ាងលម្អិត និងរកអោយឃើញនូវគ្រប់ព័ត៌មានទាំងអស់ដែលអាចផ្គត់ផ្គង់ដល់អង្គភាព។ System Analysis គឺសំដៅទៅលើវិភាគពីរយ៉ាងគឺ វិភាគស្ថានភាពបច្ចុប្បន្នរបស់អង្គភាព និង​វិភាគតម្រូវការសម្រាប់ការបង្កើតប្រព័ន្ធថ្មី។ ជំហ៊ានដំបូងគឺ Requriement Modeling ជាអ្នកកំណត់ទីកន្លែង​​ និង ពណ៌នាពីដំណាក់ការនៃពាណិជ្ជកម្ម ( Business ) ។ Requriement Modeling កំពុងបន្តការស៊ើបអង្កេតក្នុងកំឡុងពេលដែលចាប់ផ្ដើមដំបូងនូវគម្រោងប្រព័ន្ធដែលចាំបាច់ត្រូវរួមបញ្ចូលនូវបច្ចេកវិទ្យា ជាក់ស្ដែងដូចជាការសម្ភាសន៍ការស្រាវជ្រាវ ការសង្កេតការណ៍និងគំរូផ្សេងៗ។ ការងារបន្ទាប់មកទៀត គឺ Data Modeling និង Object Modeling ជាអ្នកបង្កើតនូវ Logical Model សម្រាប់ Business Process ហើយត្រូវតែស្របទៅនឹងប្រព័ន្ធដំណើរការ។

សរុបមកលទ្ធផលដែលយើងទទួលបានក្នុងដំណាក់ System Requrements Document ។ System Requriements Document គឺពិពណ៌នាអំពីការ គ្រប់គ្រងនិងតម្រូវការរបស់អ្នកប្រើប្រាស់ថែមទាំងគម្រោងនិងតម្លៃផ្សេងៗទៀត។

(​ ឯកសារនេះដក ស្រង់ចេញ​​ ពីសៀវភៅ System Analysis and 4th  Edition និពន្ធដោយ Gary B, Shelly, Thomas J, Cashman, Harry J. Rosenbaltt ទំព័រទី​ 1.20)

ក្នុងជំហ៊ាននេះយើងសិក្សាលើពីរចំណុចសំខាន់ៗគឺៈ

* Process Analysis
* Data Analysis

Process Analysis ជាយុទ្ធសាស្រ្ត (Method) ដើម្បីបំលែង Input អោយទៅជា​ Output​ ។

Process Modeling ដែលយកមកប្រើមាន៖

* Context Diagram: ក្នុងអំឡុងពេលនៃ Requirement Modeling បានធ្វើការ​ សម្ភាសន៍ចងក្រងនូវឯកសារ និងវិធីផ្សេងៗទៀតចំពោះការប្រមូលយក នូវហេតុការណ៍ អំពីរប្រព័ន្ធហើយនិងបានរៀនពីរបៀបផ្សេងៗទៀតរបស់មនុស្ស Department, Data, Processes ដើម្បីបំពេញនូវ Support Business Operation ។ ឥឡូវយើងសង្កេត Graphical Model នៃប្រព័ន្ធបព័ត៌មាន (Information System) ដោយផ្អែកលើលទ្ធផល Fact-finding ។

(​ ឯកសារនេះដកស្រង់ចេញ​​ពីសៀវភៅ System Analysis and 4th  Edition និពន្ធដោយ Gary B, Shelly, Thomas J, Cashman, Harry J. Rosenbaltt ទំព័រទី​4.6)

External Entity

External Entity

រូបភាពទី ២.២ Context Diagram

* Data Flow Diagram (DFD): ដ្យាក្រាមមួយបង្ហាញពីររបៀប​ ផ្លាស់ប្ដូរ ទិន្នន័យ តាមរយៈប្រព័ន្ធព័ត៌មាន​ ប៉ុន្តែមិនបានបង្ហាញនូវ Program Logic ឬក៏ជំហ៊ាន Processing ទេ​ ។ Data Flow Diagram បានរៀបចំបង្ហាញឡើងវិញ Logical Model ដែលបង្ហាញពីអ្វីដែលប្រព័ន្ធត្រូវធ្វើ​ និងមិនត្រូវធ្វើ
* Data Flow Diagram Symbols: ការប្រើនិមិត្តសញ្ញានៃ DFD មា​ន មូលដ្ឋានគ្រឹះ ៤ ចំណុចគឺ

Process, Data Flow, Data Store, External Entity ។

និមិត្តសញ្ញារបស់ ​​​​ ឈ្មោះនិមិត្តសញ្ញា និមិត្តសញ្ញារបស់

Gane and Sarson​​​​​​​​​​​​​​​​​​​​​ (Symbol Name)​​​​​​​​​​​​​ Yourdon

; Process

Data Flow

Data Store

External Entity

រូបទី ២.៣ និមិត្តសញ្ញា DFD

* Process Symbol: មានតួនាទីសម្រាប់ទទួលនូវការបញ្ចូលទិន្នន័យ (Input Data) និងបំលែងទិន្នន័យអោយទៅជាលទ្ធផល​ (Output Data) ។
* Data Flow Symbol: មានតួនាទីប្រាប់ជាចរន្តរត់នៃសុំណុំទិន្នន័យ (a set of data) ដែលធ្វើការផ្លាស់ប្ដូរពីផ្នែកមួយទៅផ្នែកមួយទៀត ។
* Data Store Symbol: មានតួនាទីប្រាប់ពី​ Files or tables ឬកន្លែងដែលផ្ទុកទិន្នន័យ​​ ។
* External Entity Symbol: មានតួនាទីប្រាប់ពីស្ថាប័ន អាចជាមនុស្ស ឬផ្នែកណាមួយនៃស្ថាប័នឬប្រព័ន្ធផ្សេងៗទៀតដែលបានផ្ដល់ ឬទទួលយកទិន្នន័យ ពីប្រព័ន្ធដែលវាភ្ជាប់ទៅនឹង Process ដោយ​Data Flow ។ ខាងក្រោមគឺជារូបភាពដែលបង្ហាញពីភាពត្រឹមត្រូវឬមិនត្រឹមត្រូវនៃការគូស data flow diagram តាម​​ rule ខាមក្រោម:
* គ្មាន Process ណាដែលមានតែ Output នោះទេ

Incorrect Correct

* គ្មាន Process ណាដែលមានតែ Input នោះទេ

Incorrect Correct

* Process នីមួយៗត្រូវមានឈ្មោះ
* ទិន្នន័យ​ ( Data ) មិនអាចចេញពី Data Store ទៅ Data Store មួយទៀតដោយផ្ទាល់បានទេ​ ។
* ត្រូវតែឆ្លងកាត់ Process ជាមុនសិន

Incorrect Correct

* Data មិនអាចចូលទៅកន្លែងផ្ទុកទិន្នន័យ​​ (Data Store) ដោយផ្ទាល់ពីប្រភេទខាងក្រៅនោះទេវាត្រូវឆ្លងកាត់ Process ដែលទទួល Data ពីប្រភព (Source) នោះ ។

Incorrect Correct

* ទិន្នន័យមិនអាចចេញពី​ Data Store​ ទៅ External Entity​ ដោយផ្ទាល់បានទេវាត្រូវឆ្លងកាត់តាម

រយៈ Process ។

Correct

Incorrect

* Data Store ត្រូវមានឈ្មោះ
* Data មិនអាចចេញពី​ External Entity​ មួយទៅ ដោយផ្ទាល់បានទេវាត្រូវឆ្លងកាត់តាមរយៈ​ Process ។

Correct

Incorrect

* External Entity ត្រូវមានឈ្មោះ
* Data Flow ចេញពី​ Symbol មួយទៅ ​ Symbol​ ទៀតមានទិសដៅតែមួយគត់

Correct

Incorrect

* Data Flow​​ បែកខ្នែង (Fork in Data Flow) មានន័យថា​ Data ដូចគ្នាចេញពី ទីតាំងណាមួយទៅកាន់ Process​, Data Store External Entities ពីរ​ ឬច្រើន​។

Incorrect Correct

A

A

A

B

* Data Flow ចូលរួមគ្នា (A join in a Data Flow) មានន័យថា Data ដូចគ្នាចេញពី Processes, Data store ឬ External Entity ពីរឬច្រើនផ្សេងៗគ្នាទៅកាន់ទីតាំងមួយ​ ។

Incorrect Correct

A

* Data Flow មិនអាចត្រឡប់ក្រោយទៅកាន់ Process ដដែលតែមួយនោះវិញបានទេវាត្រូវមាន Process ផ្សេងទៀតយ៉ាងហោចណាស់មួយដែលគ្រប់គ្រង data flow នេះបង្កើតជា Data Flow ផ្សេងទៀត និង ផ្ទាល់នូវ data flow ដើម្បីទៅ​ Process មុខវិញ​ ។

Incorrect Correct

* Data Flow​​ ចេញពី​ Data Store មានន័យថា​​ Read Data ពី Data Store នោះ ។
* Data Flow​ ​​ទៅ​ Data Store​ មានន័យថា​​​ Update data ក្នុង Data Store (Update data អាចជាការលុបឬការធ្វើការផ្លាស់ប្ដូរ Data)​ ។
* Data Flow​​ ត្រូវមានឈ្មោះ

២.២.៣ ការរចនាប្រព័ន្ធ (System Design):

ក្នុងដំណាក់កាលនេះគឺជាការបំលែងអ្វីដែលបានវិភាគក្នុងដំណាក់កាល Analysis ទៅជាភាសាទូទៅមួយដែលគ្រប់អ្នកសរសេរកម្មវិធី (Programmer) អាចរៀបចំការងារតាមរយៈកុំព្យូទ័រ ។ រយៈពេលនៃ System Design អត្តសញ្ញាណដែលចាំបាច់គឺ Input, Output , Interference និង​ Processes ។ បន្ថែមពីនេះត្រូវ​ Design Internal, External controls ដោយផ្អែកលើកុំព្យូទ័រនិង​​ កិច្ចការដែលធ្វើដោយដៃដែលធានាថាត្រឹមត្រូវពឹងផ្អែកទៅលើការងារនោះ។ Design នេះគឺជាឯកសារនៅក្នុង Systems Design Specification និងបង្ហាញចំពោះការគ្រប់គ្រងអ្នកប្រើប្រាស់សម្រាប់រំលឹកឡើងវិញ និងបានព្រមព្រៀងរបស់អ្នកទាំងនោះ។ (ឯកសារដកស្រង់ពីសៀវភៅ Systems Analysis and Design 4th Edition និពន្ធដោយ​ Gary B. Shelly, Thomas J. Cashman, Harry J. Rosenblatt ទំព័រទី ១.២០)

* Physical Implementation: ក្រោយពីការវិភាគទិន្នន័យរួចរាល់ហើយនោះ គម្រោងនៃទិន្នន័យជាក់ស្ដែង និងគំរូនៃដំណើរការត្រូវបានគេប្រើដូចជាមូលដ្ឋានគ្រឹះ សម្រាប់ការអនុវត្តន៍នៃ Database ។ ដំណាក់កាលសំខាន់បំផុតនៃ Physical Implementation គឺបង្កើតនៃ​Database​ មួយ ដែលអាចបត់បែនយ៉ាងលឿនក្នុងការប្រើប្រាស់ ។
* index:​ ហេតុអ្វីបានជាយើងត្រូវបង្កើត index?

​​​​​​​​​​ ​​​ index ​ជា Key ដែលបង្កើតសម្រាប់ Column ណាមួយជាក់លាក់ក្នុងគោលបំណងដើម្បបង្កើតនូវប្រសិទ្ធភាពនៃការហៅ និងស្វែងរកទិន្នន័យមកប្រើ ។ ប្រសិទ្ធភាពនៃការហៅទិន្នន័យមកប្រើ ជាកត្តាសំខាន់ក្នុងការកំណត់ Performance ។ ដោយការប្រើ index​ យើងអាចបង្កើតល្បឿនក្នុងការ Access ទៅលើ Table ដោយយើងអាចបំបាត់ចោលការត្រួតពិនិត្យមើល (Scan) តម្លៃទាំងអស់របស់ Table ។ គុណវិបត្តិនៃ Index គឺបង្កើតនូវ Performance នៃការ​ Access ទៅលើ Table ជាពិសេសនៅពេលដែលហៅទិន្នន័យពី Table ច្រើន ។

ទន្ទឹមនឹងគុណសម្បត្តិវាក៏ផ្ដល់នូវគុណវិបត្តិខ្លះៗ ដូចខាងក្រោម :

* បន្ថែមទំហំទៅលើ Database ធ្វើអោយ Update យឺតជាងពីព្រោះ Index ក៏ត្រូវ ការ Update ដែរ ។
* ចំនួននៃការ Input/Output កើតឡើងពីព្រោះតែគុណវិបត្តិទាំងនេះធ្វើអោយចំនួននៃ index ត្រូវមានកំណត់ ។

ចំណុចសំខាន់ ១ ត្រូវចងចាំដើម្បីបង្កើត Index :

* Indexes គូរបង្កើតចំពោះ​៖ Columns ណាដែលប្រើញឹកញាប់
* បង្កើតចំពោះ Columns ដែលគេប្រើសម្រាប់ជាលក្ខខ័ណ្ឌក្នុងការ Join Tables
* បង្កើតសម្រាប់ Columns ដែលផ្ទុកនូវទិន្នន័យមិនប្រែប្រួលជាជាង Columns ដែលផ្ទុកនូវទិន្នន័យប្រែប្រួល ។
* បង្កើតចំពោះ Table ដែលមាន Rows ច្រើនឬ Columns ច្រើន ។ (ឯកសារនេះដកស្រង់ចេញពីសៀវភៅ Database System Design បោះពុម្ពលើកទី១ដោយក្រុមហ៊ុន Jujisu Learning Media Limited នៅខែ មីនា ឆ្នាំ ២០០០ ទំព័រទី ១២៧ ដែលមានអាស័យដ្ឋាន ISY Nishi-kamata, Ota-ku, Tokyo 144-0051, Japan, TEL: (+81)-3-3730-4086 FAX(+81)3-3730-3642)​ ។
* Physical Data Design: គឺជាការប៉ាន់ប្រមាណមើល Hardware អោយត្រូវនឹង Database ។ ការ Design នោះត្រូវអនុវត្តន៍តាមការចោទសួរដូចខាងក្រោម:
* ក្នុង Database​ មាន Table ចំនួនប៉ុន្មាន?
* Table ​នីមួយៗមាន File ចំនួនប៉ុន្មាន? File នីមួយៗមាន Data Type ប្រភេទអ្វី? Data Type ដែលប្រើនោះត្រូវការ Space ចំនួនប៉ុន្មាន Bytes?
* Table ដែលត្រូវបញ្ចូល Data ជារៀងរាល់ថ្ងៃ (Transaction Table) មានប៉ុន្មាន? នឹងជាមធ្យមត្រូវមាន Record ប៉ុន្មាន? ដែលត្រូវបញ្ចូលក្នុង Table នោះ?
* Table ដែលត្រូវបញ្ចូល Data ដើម្បីទុកប្រើរហូត (Master Table) មាន ចំនួនប៉ុន្មាន?
* Database ជាប្រភេទ Distributed Database (Data រក្សារទុកក្នុងច្រើនកន្លែង) ឬជាប្រភេទCentralize Database (Data រក្សារទុកក្នុងកន្លែងតែមួយ) ​។
* Application Program មួយណាដែលប្រើជា Database Engine ក្រោយពីដោះស្រាយបញ្ហាតាមការចោទសួររួចហើយ ទើបគេអាចប៉ាន់ប្រមាណ មើលថាតើ Hardware ប្រភេទណាដែលអាចប្រើបាន ដូចជាទំហំ HDD, RAM, CPU ជាដើម​​​ ។

(ដកស្រង់ចេញពីសៀវភៅឯកសារ Modern database Management និពន្ធដោយ​​ Fred R. McFadden, Jeffrey A, Hoffer, Mary B, Prescott បោះពុម្ពលើកទី ៥ ដោយក្រុមហ៊ុន Pearson Education, Inc.)

២.២.៤ ការអនុវត្តន៍ប្រព័ន្ធ (System Implementation):

ការអនុវត្តន៍ប្រព័ន្ធ គឺជាការងាររបស់ System Builders ដើម្បីបំលែងពី Logical System Models អោយទៅជា Physical System ហើយការងារនេះមានដូចជាការបង្កើត Tables,​ ការ បង្កើត Form​ ។ មានន័យថាក្នុងដំណាក់កាលនេះគឺជាការងាររបស់អ្នក​ សរសេរកម្មវិធី (Programmer) ដែលត្រូវបំលែងអ្វី១ក្នុងការ Design ចូលទៅក្នុង​កុំព្យូទ័រតាមរយៈ Data Engine និង Program Language ណាមួយហើយការងារក្នុងជំហ៊ាននេះរួមមាន :

* Create Database Definition: បង្កើត Database ក្នុង​​ Database Engine
* Create Program Code: ដោយសរសេរ Coding បញ្ជាអោយធ្វើការក្នុង Programming Language ណាមួយ ។
* ដាក់ឈ្មោះអោយប្រព័ន្ធនិង​ Compile ចេញពី Programming Language
* Documentations: គឺបង្កើតជាឯកសាររបស់ប្រព័ន្ធដែលបានបង្កើតដើម្បីងាយស្រួល ក្នុងការកែប្រែ Update និងកែកំហុសនៅថ្ងៃក្រោយ​ ។

២.២.៥ ការប្រតិបត្តិ និងទ្រទ្រង់ប្រព័ន្ធ System Operation and Support:

ក្នុងជំហាននេះជាដំណាក់កាលចុងក្រោយនៃ System Development Life Cycle ហើយវាជាការថែរក្សានូវប្រព័ន្ធក្រោយពីប្រព័ន្ធនេះត្រូវដាក់អោយដំណើរការរួមទាំងការថែទាំ កម្មវិធី និងកែលំអ ប្រព័ន្ធក្នុងនោះមានដូចជា:

* Testing and Debugging The System: ធ្វើការសាកល្បងរកកំហុសក្នុងប្រព័ន្ធដែលបានបង្កើត ។
* កែតម្រូវ​ភាពខុសឆ្គង Correct Errors
* Recovery The System
* សម្រួលប្រព័ន្ធ Adapt The System To New Requirements
* Data For Testing: រកទិន្នន័យចាស់ពិតប្រាកដមកសាកល្បងប្រើប្រាស់ ។
* ជួយអ្នកប្រើប្រាស់ Assist Users Of The System មានន័យថាការបង្ហាត់បង្រៀនអ្នក ប្រើប្រាស់អោយចេះប្រើប្រព័ន្ធថ្មីនេះ ។

ជំពូក​​ ៣

វីធីសាស្រ្តនៃការសា្រវជ្រាវ

៣.១ ការរៀបចំការស្រាវជ្រាវ (Research Design)

ក្នុងការងារនីមួយៗដើម្បីឈានទៅរកភាពជោគជ័យបានគឺទាមទារអោយគេស្វែងរកនូវវិធីសាស្ត្រថាតើធ្វើដូចម្តេចដើម្បីដោះស្រាយការងារនោះអោយបានសម្រេច ។ យ៉ាងណាមិញចំពោះការងារនៃការសិក្សាអំពីប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី)​ ក្រុមយើងខ្ញុំបានជ្រើសរើសនូវវិធីសាស្រ្តមួយតាមគម្រូ System Development Life Cycle (SDLC) ដើម្បីអោយការងារនោះសម្រេចបានជារូបរាងឡើង ។ ការសិក្សាស្រាវជ្រាវដើម្បី បង្កើតជាប្រព័ន្ធថ្មីនេះបានអនុវត្តន៍ តាមជំហានទាំងឡាយដូចមានក្នុងដំណាក់កាលដូចខាងក្រោមៈ

ជំហានទី១: ពិភាក្សា និងសិក្សាទៅលើគម្រោងការងារ

(Discussion, System Planning)

ជំហានទី២: ធ្វើសម្ភាស៍ ប្រមូលទិន្នន័យ និងរៀបចំឯកសារ

​​ (Interview, Data Collection And Documentation)

ជំហានទី៣: ការសិក្សាអំពីតម្រូវការប្រព័ន្ធថ្មី (Requirement)

ជំហានទី៤: វិភាគវែកញែករកវិធីដោះស្រាយ (System Analysis)

ជំហានទី៥: បង្កើតប្រព័ន្ធថ្មី និង​ សរសេរកូដ

(System Implementation)

ជំហានទី៦: **ការដាក់ប្រព័ន្ធឱ្យដំណើរការ និងថែទាំប្រព័ន្ធ**

**(System Operation And Support)**

៣.១.១ ជំហានទី១ ពិភាក្សា និងសិក្សាទៅលើគំរោងការងារ

ក្នុងជំហាននេះក្រុមយើងខ្ញុំបានធ្វើការសិក្សាទៅលើគម្រោងការងារ ដើម្បីកំណត់លើចំណុចធំដែល ជាគោលដៅចម្បងរបស់ប្រព័ន្ធនេះ ដើម្បីយកមកធ្វើការចុះស្រាវជ្រាវអោយបានល្អិតល្អន់ និងម្យ៉ាងទៀតក្រុម ការងារយើងខ្ញុំបានស្វែងរកឯកសារទាំងឡាយដែលមានលក្ខណៈប្រហាក់ប្រហែលនឹងប្រព័ន្ធនេះ សម្រាប់ការ ពិភាក្សា និងបង្កើតជាគំនិតថ្មី១ក្នុងការចុះស្រាវជ្រាវ ។

៣.១.២ ជំហានទី២ ធ្វើសម្ភាសន៍ ប្រមូលទិន្នន័យ និងរៀបចំឯកសារ

ក្នុងជំហាននេះគឺការចុះទៅ និងពិនិត្យមើលដំណើរការរាល់បញ្ហាការងារដោយផ្ទាល់នៅ វិទ្យាស្ថានពហុបច្ចេកទេសភូមិភាគតេជោសែនបាត់ដំបងនិងចុះពិនិត្យផលពិបាកការងារដែលយើងត្រូវយកមកសិក្សាការងារជាមួយនិងបុគ្គលិកហើយសាកសួរពីដំណើរការងារផ្សេងៗដែលពាក់ព័ន្ធ ដើម្បីធ្វើការប្រមូល ព័ត៌មាន របស់ប្រព័ន្ធការងារបច្ចុប្បន្ន ថាតើប្រព័ន្ធការងារនេះត្រូវបញ្ចូល Data របៀបណា ផ្តល់ Report ទាំង នោះត្រូវផ្ដល់ទៅផ្នែកណាខ្លះ? និងប្រមូលឯកសារ ដែលមានស្រាប់ដូចជាឯកសារដែលពាក់ព័ន្ធទៅនឹង ទិន្នន័យរបស់គ្រូបង្រៀន និងអ្នកកាន់បញ្ជី ជាពិសេសពាក្យចុះឈ្មោះចូលរៀនរបស់សិស្ស គឺជាតម្រូវការរបស់ ប្រព័ន្ធគ្រប់គ្រងថ្មីដែលនឹងត្រូវធ្វើការបង្កើតដើម្បីអោយមានភាពងាយស្រួល ក្នុងការគ្រប់គ្រងនិងចំណេញពេលវេលា ។

* ប្រមូលរាល់ Data ទាំងអស់ដែលទាក់ទងប្រព័ន្ធ
* រៀបចំលទ្ធផលនៃ Interview And Data Collection
* រៀបចំការណាត់ជួបអ្នកប្រើប្រាស់ប្រព័ន្ធ
* រៀបចំសំនួរសម្រាប់សួរអ្នកប្រើប្រាស់ប្រព័ន្ធ
* កត់ត្រារាល់ចម្លើយរបស់អ្នកប្រើប្រាស់
* កត់ត្រាតម្រូវការបន្ថែម ពីអ្នកប្រើប្រាស់

៣.១.៣ ជំហានទី៣ ការសិក្សាអំពីតម្រូវការប្រព័ន្ធថ្មី

ដំណាក់កាលដែលត្រូវចុះធ្វើការស្រាវជ្រាវនៅវិទ្យាស្ថានពហុបច្ចេកទេសភូមិភាគតេជោសែនបាត់ដំបង ដោយផ្ទាល់ដោយការសាកសួរអំពីព័ត៌មានលំអិតនៅតាមការិយាល័យដែលពាក់ព័ន្ធ ដើម្បីយកមកវិភាគ ធ្វើការរៀបចំ និងកែតម្រូវឡើងវិញ។ ម្យ៉ាងទៀតក្រុមការងារយើងខ្ញុំបានចុះមកអង្កេតការជាមួយ និងលោកគូ អ្នកគ្រូដែលកាន់កម្មវិធីនេះដើម្បីងាយស្រួលនិងយល់កាន់តែច្បាស់ទៅលើដំណើរការនៃប្រព័ន្ធដែលបង្កភាពងាយស្រួលដល់ការរៀបចំ និងគ្រប់គ្រងទៅលើប្រព័ន្ធថ្មីអោយកាន់តែមានប្រសិទ្ធភាពឡើង ។

៣.១.៤ ជំហានទី៤ វិភាគវែកញែករកវិធីដោះស្រាយ

ជំហាននេះទាក់ទងនិងជំហានខាងលើគឺពេលប្រមូល Data And Information បានហើយត្រូវ ធ្វើ ការបំលែងពីប្រព័ន្ធដែលកំពុងប្រើប្រាស់ពីមុនទៅប្រព័ន្ធថ្មីមួយទៀតដែលអ្វីៗ ត្រូវដោះស្រាយដោយកុំព្យូទ័រស្ទើរតែទាំងអស់ និងបន្ថែមនូវអ្វីដែលមានសារៈប្រយោជន៍លើប្រព័ន្ធចាស់ ។ វិភាគអោយដឹងថាត្រូវធ្វើបែបណា ទើបចេញជា Information ត្រឹមត្រូវនិងបំលែងការគណនានោះតាមរយះ Program Code ។ ជំហាននេះរួមមាន Data Flow ការផ្ទុក Data ក្នុង Diagram System ដែលតំណាងអោយ សកម្មភាពរបស់ប្រព័ន្ធការងារទាំងមូលលើសពីនេះទៅទៀតជំហាននេះក៏បានរៀបចំផងដែរនូវអ្វីៗ ដែលប្រមូលបានពីជំហានមុខៗមកបំលែងអោយមានទម្រង់ជា

Input Processes Output

របស់ប្រព័ន្ធថ្មីតាមរយៈកុំព្យូទ័រដែលរហ័សបំផុត​ ។ ក្នុងជំហាននេះបើជួបនូវបញ្ហានោះដើម្បីជៀសវាងនូវកំហុសឆ្គងដែលបណ្ដាលមកពីព័ត៌មាន មិនច្បាស់ធ្វើអោយខាតបង់ពេលវេលាក្នុងការប្រែឡើង វិញនូវដំណាក់កាលក្រោយៗបន្តបន្ទាប់ទៀតត្រូចត្រឡប់ទៅជំហានទី១ម្ដងទៀត ។

៣.១.៥ ជំហានទី៥: បង្កើតប្រព័ន្ធថ្មី និង សរសេរកូដ

ក្រោយពីធ្វើការវិភាគ ក្នុងជំហាន Analysis​ គ្រប់អ្វីៗដែលបានធ្វើក្នុងជំហានទី៣ ត្រូវបាន បំលែងឡើងវិញអោយមានលក្ខណៈ Computer Program Design Method ដែលគ្រប់ Programmers អាចធ្វើការរៀបចំសរសេរបញ្ហាទៅកុំព្យូទ័រតាមរយៈ Program Language ដែលសមស្រប ។ ក្នុងនោះរួមមាន​ Entity Relational Diagram, Flowchart, Data Dictionary, Input Output Design និងរៀបចំសរសេរ Code ដើម្បីអោយ Database ដែលរៀបចំបង្កើត ថ្មីក្លាយជាប្រព័ន្ធគ្រប់គ្រងមួយមានលក្ខណៈពេញលេញ ។ ប៉ុន្តែបើសិនជានៅមានបញ្ហា កើតឡើងនៅ ក្នុងពេលអនុវត្តន៍ Code ឬ Database ត្រូវតែលុប ទៅធ្វើការជំហានទី៤ឡើងវិញ ។

៣.១.៦ ជំហានទី៦ ដំណើរការនៃប្រព័ន្ធ និងថែទាំប្រព័ន្ធ

ជាជំហានចុងក្រោយដែលធ្វើការដាក់អោយប្រើប្រាស់ និងធ្វើការថែទាំនូវប្រព័ន្ធថ្មីនេះ ដើម្បីជៀសវាងមានបញ្ហា error កើតឡើង ។ ប្រព័ន្ធនេះក្លាយទៅជាប្រព័ន្ធគ្រប់គ្រងថ្មី ដែលបានបង្កើតឡើង អោយមានលក្ខណៈចេញជារូបរាងហើយនោះចាំបាច់ត្រូវធ្វើការសាកល្បងជាមុនសិន ប្រសិនបើប្រព័ន្ធ នៃការគ្រប់គ្រងថ្មីមានលក្ខណៈត្រឹមត្រូវ ហើយនោះទើបដាក់អោយ End User ប្រើប្រាស់ព្រមទាំងបង្ហាញនូវ របៀបប្រើប្រាស់ និងរង់ចាំទទួលមកវិញផងដែរ បន្ទាប់ពីការសាកល្បងប្រើប្រាស់នៃ End User ជួបនូវបញ្ហាផ្សេងៗដែលកើតឡើងដោយស្មានមិនដល់ ឬដោយ Program ធ្វើមិនបាន ។

៣.២ វិធីសាស្រ្តស្រាវជ្រាវ (Research Methodology)

ដូចយើងបានដឹងហើយថា ក្នុងដំណាក់កាលនេះតម្រូវការនៃប្រព័ន្ធ គឺត្រូវប្រមូលទិន្នន័យដែលពាក់ព័ន្ធទៅនឹង System ។ ដើម្បីប្រមូលទិន្នន័យអោយបានត្រឹមត្រូវ និងមានប្រសិទ្ធភាពខ្ពស់នោះយើងត្រូវប្រើវិធីសាស្រ្តមួយដែលគេឱ្យឈ្មោះថាបច្ចេកវិទ្យាស្វែងរកការពិត (Fact-Finding Technique)

បច្ចេកវិទ្យាស្វែងរកការពិតរួមមានៈ

* ឯកសារគម្រូ (Sampling Document)
* ការសម្ភាសន៍ (Interviewing)
* ការប្រើប្រាស់សំណួរ (Using Questionnaires)
* ការសង្កេត (Observing)

៣.២.១ ប្រភេទទិន្នន័យ

ប្រភេទទិន្នន័យត្រូវបានបែងចែកជាពីរ

* ទិន្នន័យចម្បង (Primary Data)
* ជាទិន្នន័យដើមចាំបាច់ដែលទទួលបានពីការចុះធ្វើកម្មសិក្សា
* ការសាកសួរព័ត៌មានជាមួយនឹងបុគ្គលិក នៃការិយាល័យរដ្ឋបាល
* ទិន្នន័យបន្ទាប់ (Secondary Data)
* ជាទិន្នន័យដែលបានមកពីដំណើរការ (Process) ណាមួយក្នុងប្រព័ន្ធសម្រាប់យកទៅប្រើប្រាស់បន្តក្នុងដំណើរការ (Process) មួយផ្សេងទៀត ។

៣.២.២ វិធីសាស្រ្តប្រមូលទិន្នន័យ

ក្នុងនោះផងដែរ​ ក្រុមនិស្សិតយើងខ្ញុំក៏បានចុះសម្ភាសន៍ជាមួយបុគ្គលិកផ្នែកការិយាល័យរដ្ឋបាល ​​​​​​​​​​ហើយធ្វើការ ប្រមូលរាល់ទិន្នន័យដែលទាក់ទងទៅនឹងការសាងសង់ប្រព័ន្ធថ្មី ដោយអនុវត្តន៍ដូចតទៅ៖

* ធ្វើការណាត់ជួបជាមួយអ្នកត្រូវសម្ភាសន៍ ហើយប្រាប់ពីគោលបំណងនៃការសម្ភាសន៍។
* បង្កើតនូវ List សំណួរសម្រាប់សួរ ។
* ដោយផ្តើមនូវគោលបំណងសម្រាប់សម្ភាសន៍ម្តងទៀត។
* ត្រង់ចំណុចដែលមិនយល់ត្រូវសុំការអនុញ្ញាតអោយគាត់បញ្ជាក់ម្តងទៀត។
* ជាចុងបញ្ចប់ត្រូវសង្ខេបនូវព័ត៌មាន ដែលប្រមូលបានពីការសម្ភាសន៍ និងធ្វើការបង្ហាញនូវព័ត៌មានទាំងនោះទៅកាន់អ្នកដែលត្រូវបានសម្ភាសន៍វិញ។
* ស្វែងរក ឯកសារស្រាវជ្រាវនៅបណ្ណាល័យដើម្បីបំពេញនូវចំណុចខ្វះខាត

៣.៣ ការវិភាគទិន្នន័យ

ក្នុងការវិភាគប្រព័ន្ធជាដំបូងយើងត្រូវធ្វើការសិក្សាស្វែងយល់ អំពីរាល់បញ្ហាដែលកើតមាននៅលើប្រព័ន្ធចាស់ ។

នៅក្នុងការវិភាគប្រព័ន្ធជាទូទៅត្រូវបានធ្វើឡើងតាមបីជំហានដូចខាងក្រោមៈ

* ការវិភាគទៅលើប្រព័ន្ធចាស់
* សិក្សាតម្រូវការនៃប្រព័ន្ធ
* វិភាគលើបញ្ហា
* ការវិភាគទៅលើប្រព័ន្ធចាស់

ជាការសិក្សាលើប្រព័ន្ធចាស់គឺគ្រប់សមាសធាតុទាំងអស់ដើម្បីធ្វើការកែតម្រូវរាល់បញ្ហាដែលកើតឡើងចំពោះប្រព័ន្ធចាស់ទាំងអស់មាន

* សិក្សាលើបុគ្គលិកដែលពាក់ព័ន្ធ រួមមានអ្នកប្រើប្រាស់ដើម្បីធ្វើការសាកសួររាល់បញ្ហាដែលបានជួបប្រទះ ។
* សិក្សាពីព័ត៌មាន ដែលទទួលបានដើម្បីស្វែងយល់ពីបញ្ហា
* សិក្សាពីរាល់បញ្ហាទិន្នន័យដែលធ្លាប់កើតមានក្នុងប្រព័ន្ធចាស់
* សិក្សាលើការផ្ទុកទិន្នន័យនៃប្រព័ន្ធ
* សិក្សាពីប្រភេទទិន្នន័យ ថាតើយើងគួរជ្រើសរើសយក Database ប្រភេទណា
* សិក្សាស្វែងយល់ពីដំណើការ
* សិក្សាលើតម្រូវការនៃប្រព័ន្ធ

ក្នុងការសិក្សាលើតម្រូវការនៃប្រព័ន្ធ គឺតម្រូវឱ្យយើងស្វែងយល់ឱ្យបានល្អិតល្អន់បំផុតដើម្បីផ្តល់នូវភាពងាយស្រួលមិនស៊ាំញ៉ាំ មិនស្មុកស្មាញ មិនប្រើពេលវេលាយូរ ។ ក្នុងការសង់ប្រពន្ធ័គ្រប់គ្រងមួយដែលពេញលេញ វាឆ្លើយតបទៅនឹងតម្រូវការរបស់អ្នកប្រើប្រាស់ ។ ដូច្នេះក្នុងករណីនេះអ្វីដែលយើងសិក្សារួមមានៈ

* សិក្សាលើតម្រូវការបន្ថែមពីបុគ្គលិកដែលពាក់ព័ន្ធ ថាតើគាត់ចង់បានអ្វីកែប្រែបន្ថែមទៀត ។
* សិក្សាលើតម្រូវការទៅលើការផ្ទុកទិន្នន័យ
* សិក្សាលើដំណើរការរបស់ប្រព័ន្ធ
* វិភាគលើបញ្ហា

ការវិភាគលើបញ្ហាទាំងអស់ដែលកើតឡើងក្នុងប្រព័ន្ធចាស់ និងវិភាគលើតម្រូវការបន្ថែមដោយលើកយកបញ្ហាដែលធ្លាប់ជួបប្រទះក្នុងប្រព័ន្ធការងារចាស់ និងវិភាគទៅលើព័ត៌មានដែលទទួលបានក្នុងគោលបំណងសិក្សាវិភាគអំពីបញ្ហាដែលនឹងអាចកើតមានក្នុងប្រព័ន្ធជាមុនសិន ដើម្បីជាប្រយោជន៍ក្នុងការស្វែងរកដំណោះស្រាយដែលសមស្រប និងគួរឱ្យទុកចិត្តបានធានាថាគ្មានអ្វីប៉ះពាល់អវិជ្ជមាន ។

៣.៤ លទ្ធផលនៃការវិភាគ

តាមការសិក្សាលើប្រព័ន្ធចាស់ដែលដំណើរការសព្វថ្ងៃលើផ្នែកទិន្នន័យ​ (Data) ព័ត៌មាន (Information)​ និងដំណើរការ (Process) នោះប្រព័ន្ធបង្ហាញដូចខាងក្រោម:

៣.៤.១ លទ្ធផលនៃការសិក្សាលើប្រព័ន្ធចាស់

ទិន្នន័យ (Data)​ ក្នុងប្រព័ន្ធមានដូចខាងក្រោមៈ

* ព័ត៌មានរបស់សិស្ស
* ប្រវត្តិរូបសង្ខេបរបស់សិស្ស
* មុខវិជ្ជាសិក្សា
* ម៉ោងសិក្សា
* បន្ទប់សិក្សា
* ព័ត៌មានរបស់គ្រូបង្រៀន
* ប្រវត្តិរូបសង្ខេបរបស់គ្រូបង្រៀន
* មុខវិជ្ជាបង្រៀន
* ម៉ោងសិក្សា
* បន្ទប់បង្រៀន
* ដំណើរការប្រព័ន្ធទាំងមូលក្នុងការរក្សាទិន្នន័យ

នៅពេលមានសិស្សថ្មីចូលរៀន ប្រធានការិយាល័យបានចាប់ផ្ដើមចុះឈ្មោះរបស់សិស្សដែលចូលរៀននិងព័ត៌មានផ្សេងៗ រាល់ទិន្នន័យរបស់សិស្សទាំងអស់នេះត្រូវបានកត់ត្រានិងរក្សាទុកក្នុងកុំព្យូទ័រ ដោយប្រើកម្មវិធី Microsoft Office (Excel) ។

៣.៤.២ លទ្ធផលនៃដំណោះស្រាយ

ក្រុមនិស្សិតយើងខ្ញុំបានបង្កើតនៅប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី) តាមបច្ចេកវិទ្យាព័ត៌មានវិទ្យាទំនើបដែលចែកចេញជាពីរផ្នែកនិងមានតួនាទីដូចខាងក្រោមៈ

* Font End: ជាផ្ទាំងការងារសម្រាប់អ្នកប្រើប្រាស់ប្រព័ន្ធ (User Interface) និងបង្កើតរបាយការណ៍ (Report) ដោយប្រើកម្មវិធីបង្កើតឡើងតាមរយៈកម្មវិធី Microsoft Access 2016 ។
* Back End: ជាការបង្កើតប្រព័ន្ធផ្ទុកទិន្នន័យដែលមានលក្ខណៈជាប្រព័ន្ធ Database ដែលពីមុនផ្ទុកក្នុង Microsoft Office (Excel) ដោយធ្វើការបង្វែរមកប្រើកម្មវិធី Microsoft Access 2016 ។

៣.៤.៣ ការគ្រោងនៃប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី)

* (Short Course Management System)
* អ្វីទៅដែលហៅថា Short Course Management System?​ តាមរយៈនៃការសិក្សា និងការស្វែងយល់បន្ថែមពីការអានសៀវភៅស្ដីអំពី System Design យើងអាចនឹងឱ្យនិយមន័យនៃ Short Course Management System បានដូចខាងក្រោម ។
* Database System Design នៃប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី) គឺជាការគិតនិងកំណត់ទៅលើ Logical Function ហើយនិងលក្ខណៈពិសេស​នៃប្រព័ន្ធជាពិសេសគឺបង្ហាញ បកស្រាយអំពីទំនាក់ទំនង (Relationship) នៃបណ្តា External Entity ការ Process of Data ការរក្សាទុកនូវData (Store Data) ដែលនឹងបកស្រាយតាមរយៈ Context Diagram DFD (Data Flow Diagram) នោះ ។
* ជាសង្ខេប System Database Design គឺគេចង់ថា តើប្រព័ន្ធដែលអ្នកសិក្សាកំពុងតែសិក្សាធ្វើអំពីអ្វី? ជាមួយនឹងផ្នែកណាខ្លះ?

Student Register

Request Report

Officer

Registered

System

Student

Report Info

រូបភាព ទី​ ៣.៤ System Database Design

ប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាខាងលើជា Process តំណាងអោយការងារនៃប្រព័ន្ធទាំងមូលដែលមានទំនាក់ទំនងជាមួយនឹង External Entity ចំនួនមួយគឺ

Student ជាអ្នកដែលផ្ដល់នូវព័ត៌មាន ដែលត្រូវកត់ត្រាចូលទៅក្នុង Database ។

Officer ជា Entity ដែលធ្វើការដាក់នូវសំណើរសុំចុះឈ្មោះសិស្សថ្មី ។

៣.៤.៤ ការគ្រោងនៃ Physical Database សម្រាប់ប្រព័ន្ធថ្មី

(Physical Database Design for New System)

គឺជាការបង្ហាញនៃការអនុវត្តន៍ជាក់ស្ដែងនៃប្រព័ន្ធទិន្នន័យទៅលើដំណើរការ ឬគម្រោងរបស់ Logical Design ខាងលើ។ ប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី) គឺជាឧទាហរណ៍ជាក់ស្ដែងដូចជាវិធីសាស្រ្តនៃការបញ្ចូល និងរក្សាទុកទិន្នន័យ ទម្រង់នៃការបង្ហាញរបាយការណ៍ទិន្នន័យ (Report) ។ល។ រួមមក Physical Database Design ការបកស្រាយនូវអ្វីដែលយើងអាចមើលឃើញជាក់ស្ដែង​ ។

៣.៤.៥ លទ្ធផលនៃការរចនាប្រព័ន្ធ

ដ្យាក្រាមបដិទ(Context Diagram) ជាដ្យាក្រាមបង្ហាញពីដំណើរការ​ ប្រព័ន្ធគ្រប់គ្រងមួយ និងព្រំដែននៃដំណើរការការងារ ។

Short Course

Management

System

Student Info

Request

Manager

Student Register

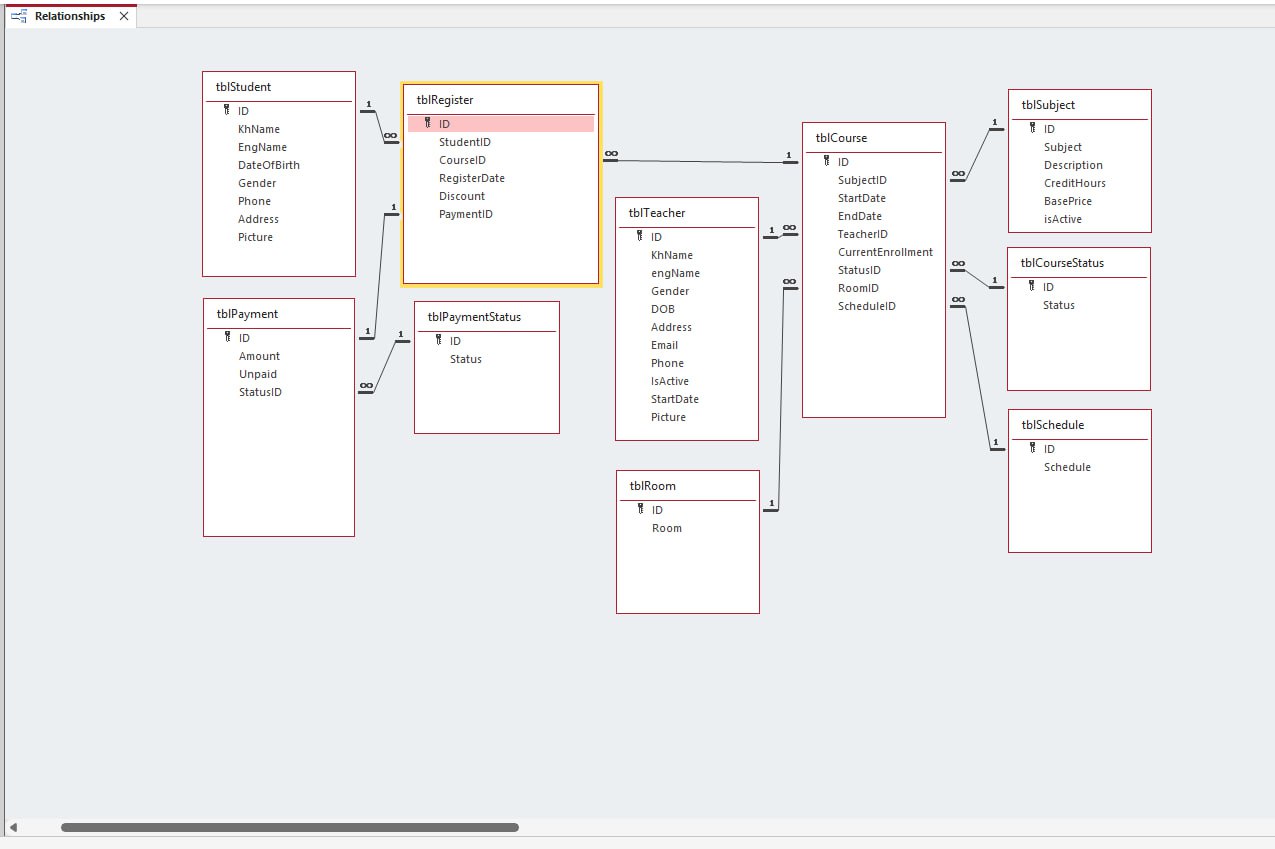
Reporting

Input Information

Operator

រូបភាព ទី ៣.៥ Short Course Management System

ទំនាក់ទំនង Table (Relational Diagram) ជាទំនាក់ទំនងរវាងTableក្នុងប្រព័ន្ធថ្មី



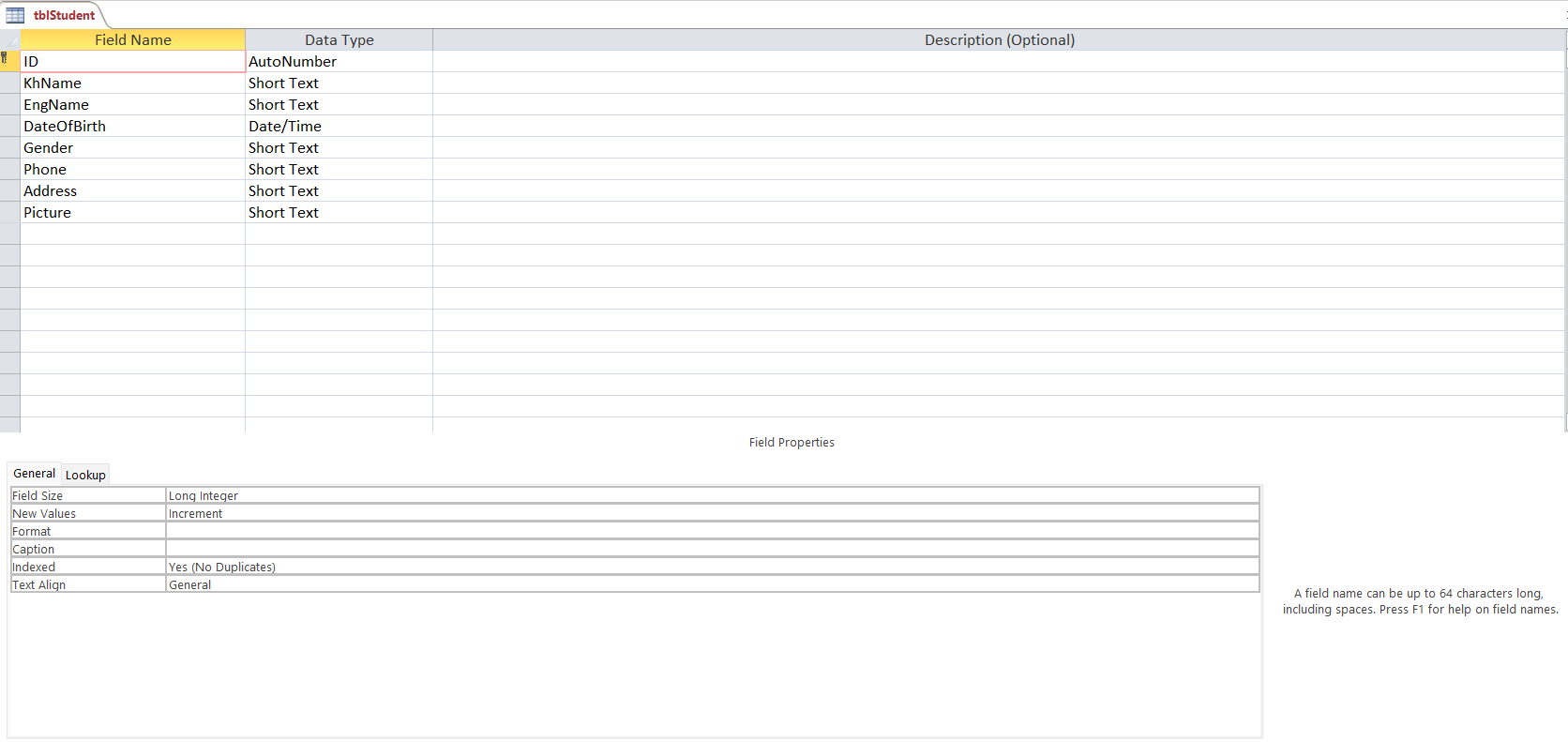
៣.៥ Data Dictionary For Table

នៅពេលនេះយើងនឹងបង្ហាញនូវ Table ដែលបានបង្កើតសម្រាប់ប្រព័ន្ធគ្រប់គ្រងថ្មី ។ ការបង្ហាញនេះ ជាលក្ខណៈមួយដែលគេហៅថា “Data Dictionary” ។ Data Dictionary នេះហើយជាមូលដ្ឋានសំខាន់សម្រាប់យកទៅបង្កើត ដេតាបេស​ (Database) ពិតប្រាកដនៅក្នុងម៉ាស៊ីន ។

៣.៥.១ Table Student

នៅក្នុង Table Student មានព័ត៌មានដូចជា

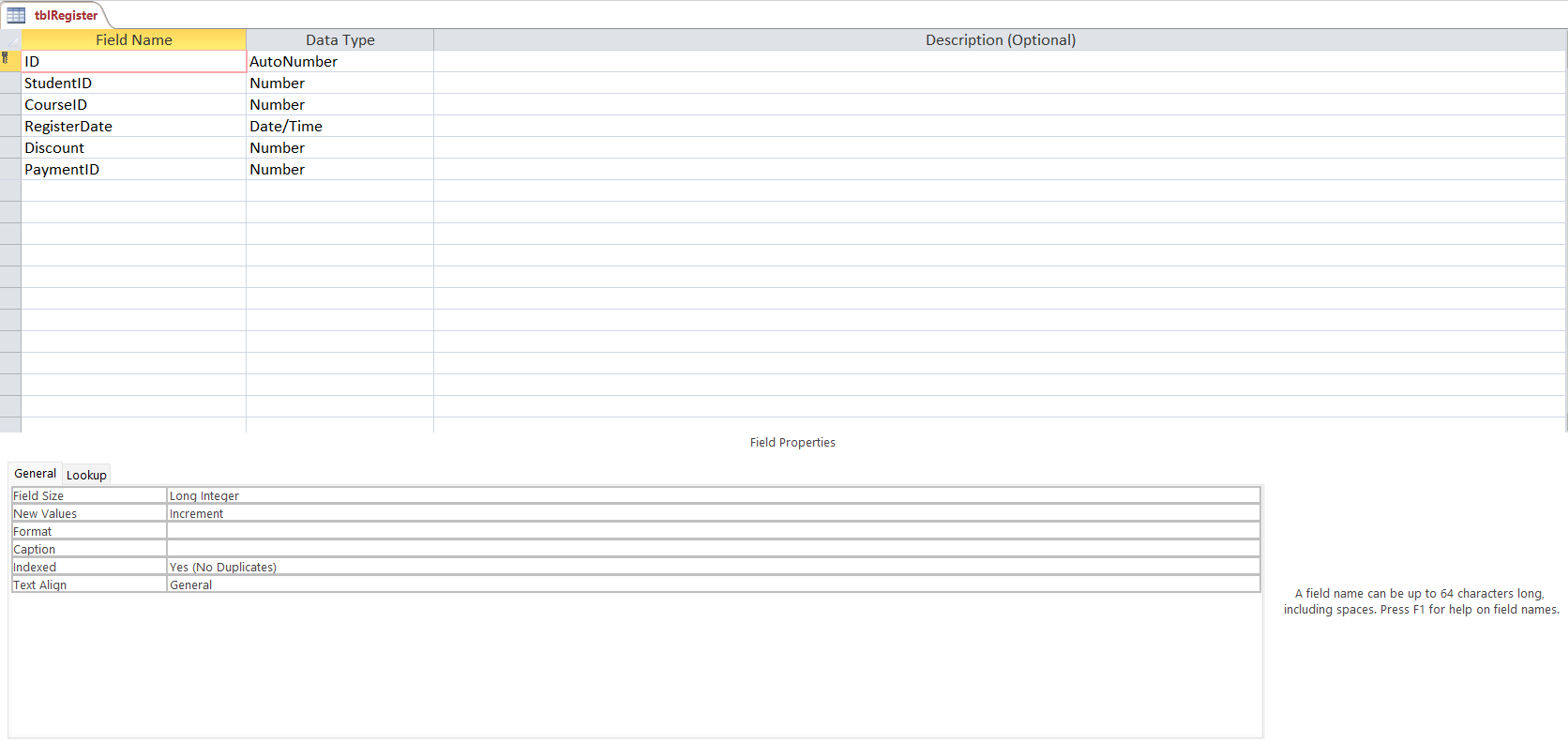
* ID : លេខសម្គាល់
* KhName :​​ ឈ្មោះជាអក្សរខ្មែរ
* EngName : ឈ្មោះជាអក្សរឡាតាំង
* DateofBirth : ថ្ងៃខែឆ្នាំកំណើតសិស្ស
* Gender :​ ភេទ
* Phone : លេខទូរស័ព្ទសិស្ស
* Address : អាស័យដ្ឋានសិស្ស
* Picture : រូបថតកាតសិស្ស



៣.៥.២ Table Register

នៅក្នុង Table Register មានព័ត៌មានដូចជា

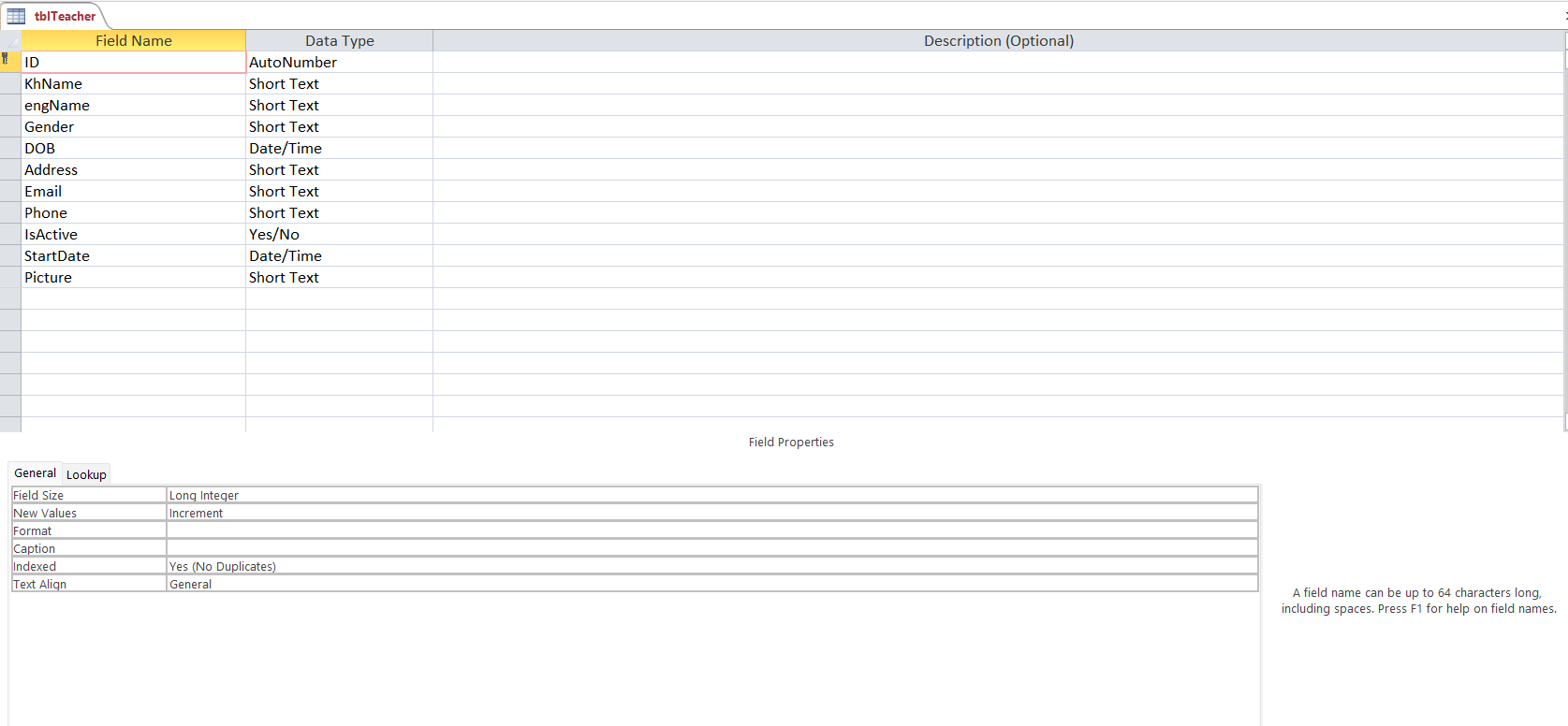
* ID : លេខសម្គាល់
* StudentID : លេខសម្គាល់សិស្ស
* CourseID : លេខសម្គាល់វគ្គសិក្សា
* RegisterDate : ថ្ងៃខែឆ្នាំចុះឈ្មោះ
* Discount : បញ្ចុះតម្លៃ
* PaymentID : លេខសម្គាល់ការបង់ប្រាក់



៣.៥.៣ Table Teacher

នៅក្នុង Table Teacher មានព័ត៌មានដូចជា

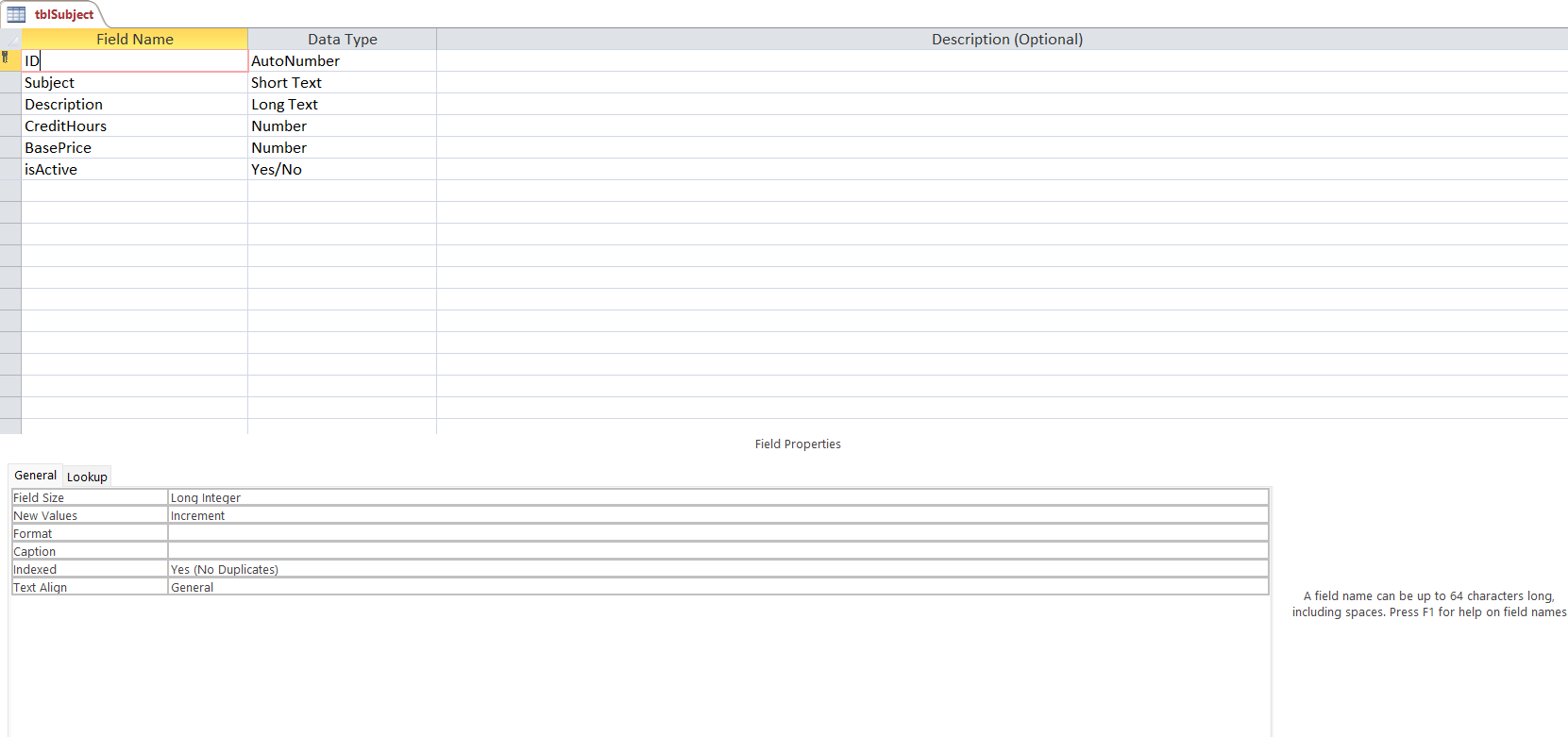
* ID : លេខសម្គាល់
* KhName : ឈ្មោះជាអក្សរខ្មែរ
* engName : ឈ្មោះជាអក្សរឡាតាំង
* Gender : ភេទ
* DOB : ថ្ងៃខែឆ្នាំកំណើតគ្រូបង្រៀន
* Address : អាស័យដ្ឋានគ្រូបង្រៀន
* Email : អ៊ីម៊ែលគ្រូបង្រៀន
* Phone : លេខទូរស័ព្ទគ្រូបង្រៀន
* IsActive : សកម្ម
* StartDate : ថ្ងៃខែឆ្នាំចាប់ផ្ដើម
* Picture : រូបថតកាតគ្រូ



៣.៥.៤ Table Subject

នៅក្នុង Table Subject មានព័ត៌មានដូចជា

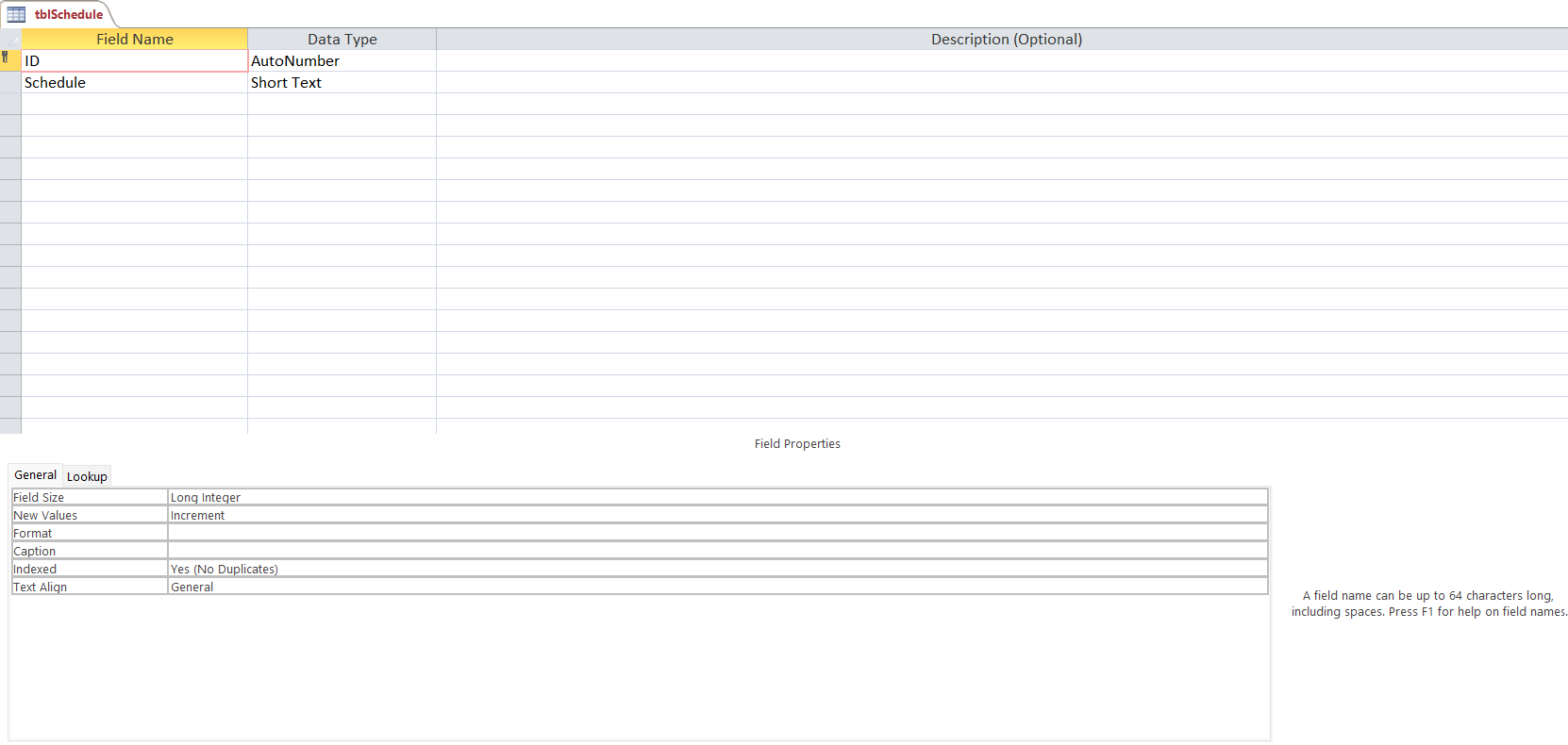
* ID : លេខសម្គាល់
* Subject : មុខវិជ្ជា
* Description : ការពិពណ៌នា
* CreditHours : ម៉ោងឥណទាន
* BasePrice : តម្លៃជាមូលដ្ឋាន
* isActive : សកម្ម



៣.៥.៥ Table Schedule

នៅក្នុង Table Schedule មានព័ត៌មានដូចជា

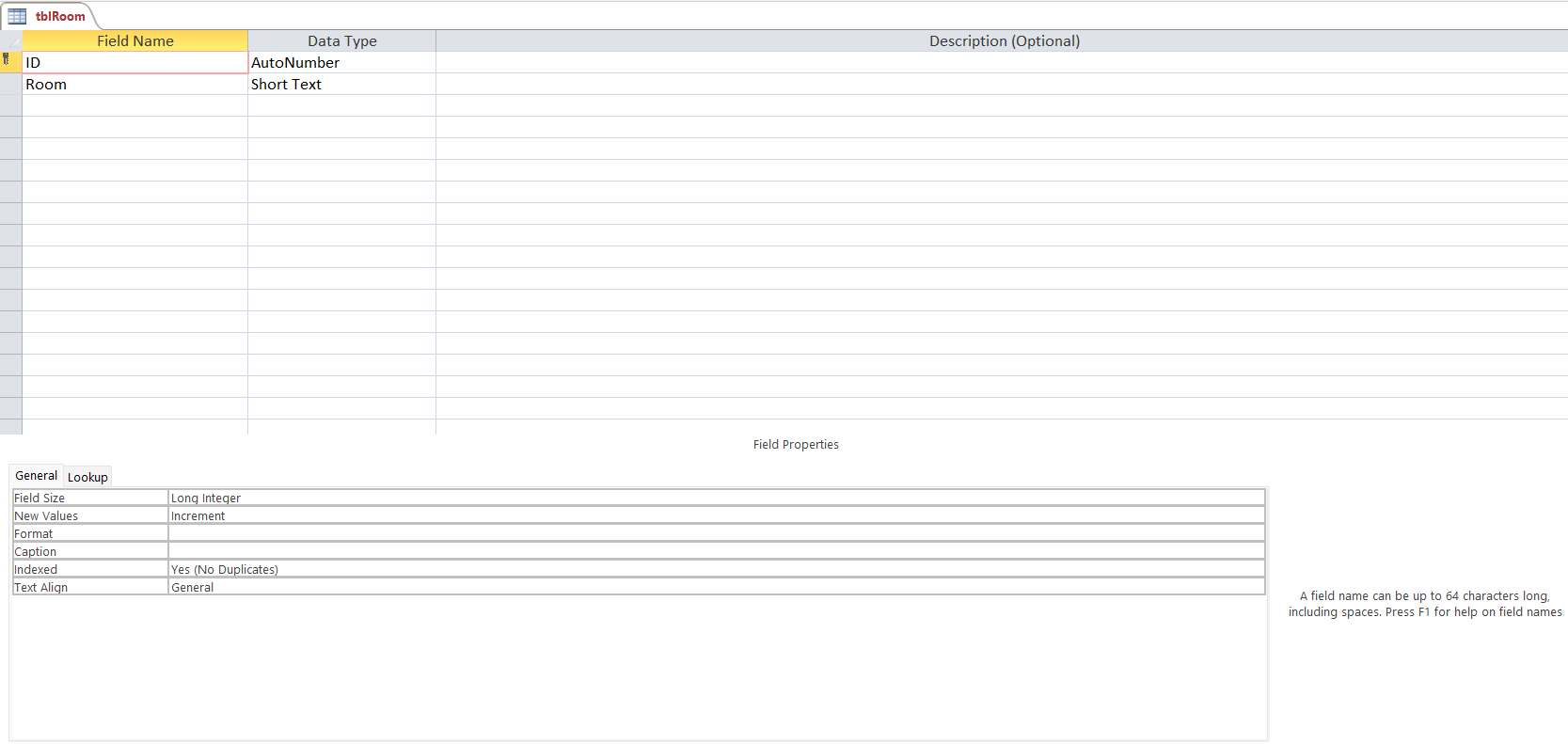
* ID : លេខសម្គាល់
* Schedule : កាលវិភាគ



៣.៥.៦ Table Room

នៅក្នុង Table Room មានព័ត៌មានដូចជា

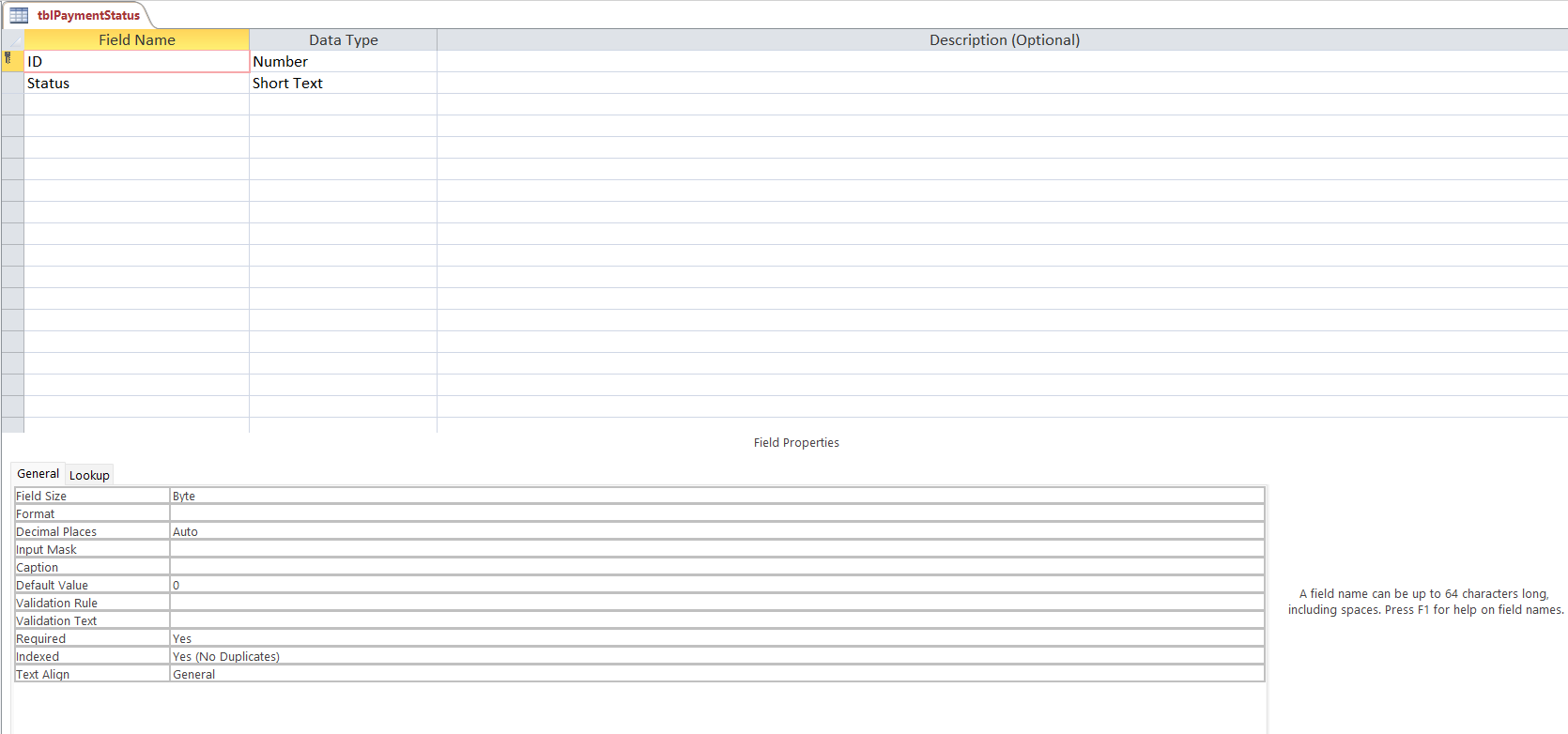
* ID : លេខសម្គាល់
* Room : បន្ទប់រៀន



៣.៥.៧ Table PaymentStatus

នៅក្នុង Table PaymentStatus មានព័ត៌មានដូចជា

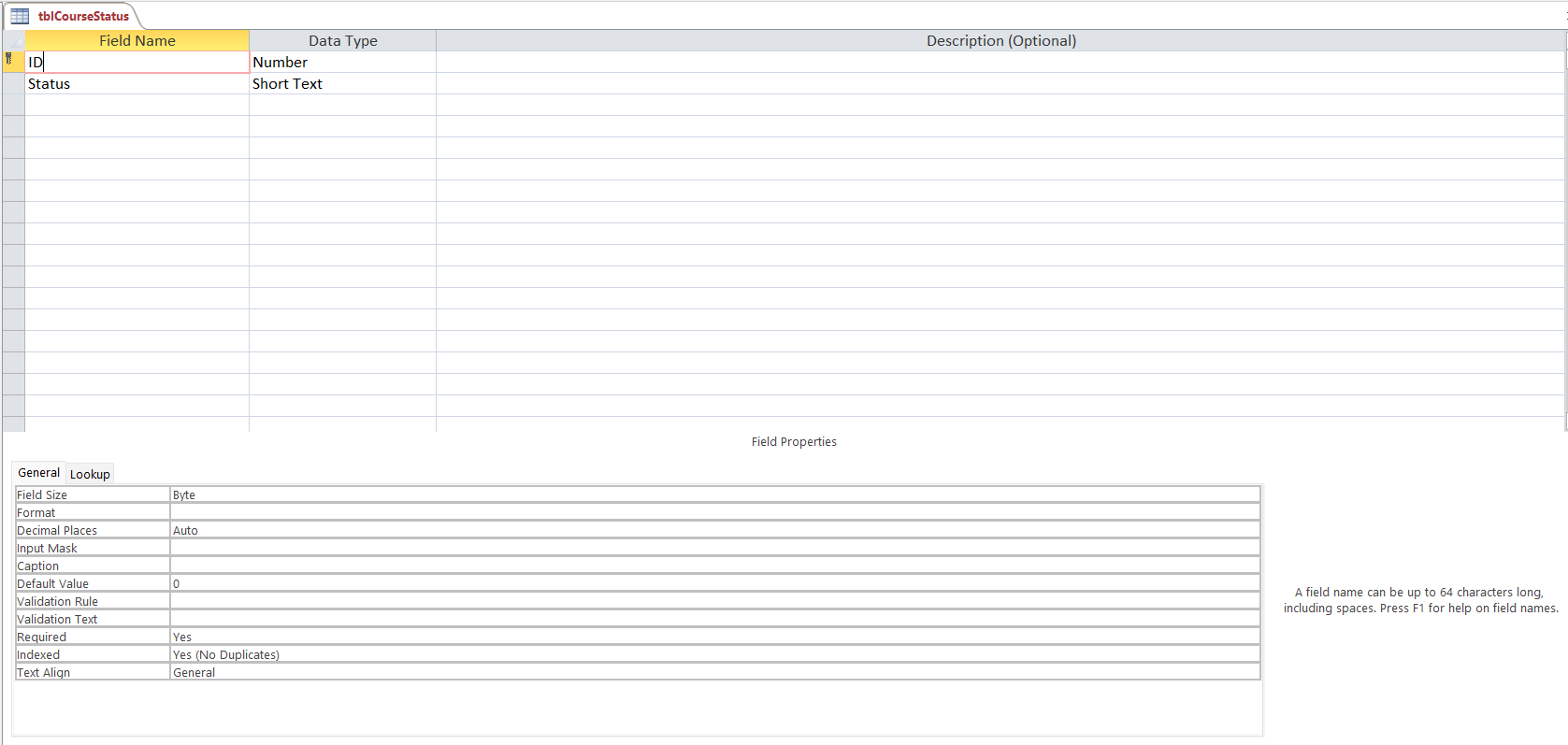
* ID : លេខសម្គាល់
* Status : ស្ថានភាពបង់ប្រាក់



៣.៥.៨ Table CourseStatus

នៅក្នុង Table CourseStatus មានព័ត៌មានដូចជា

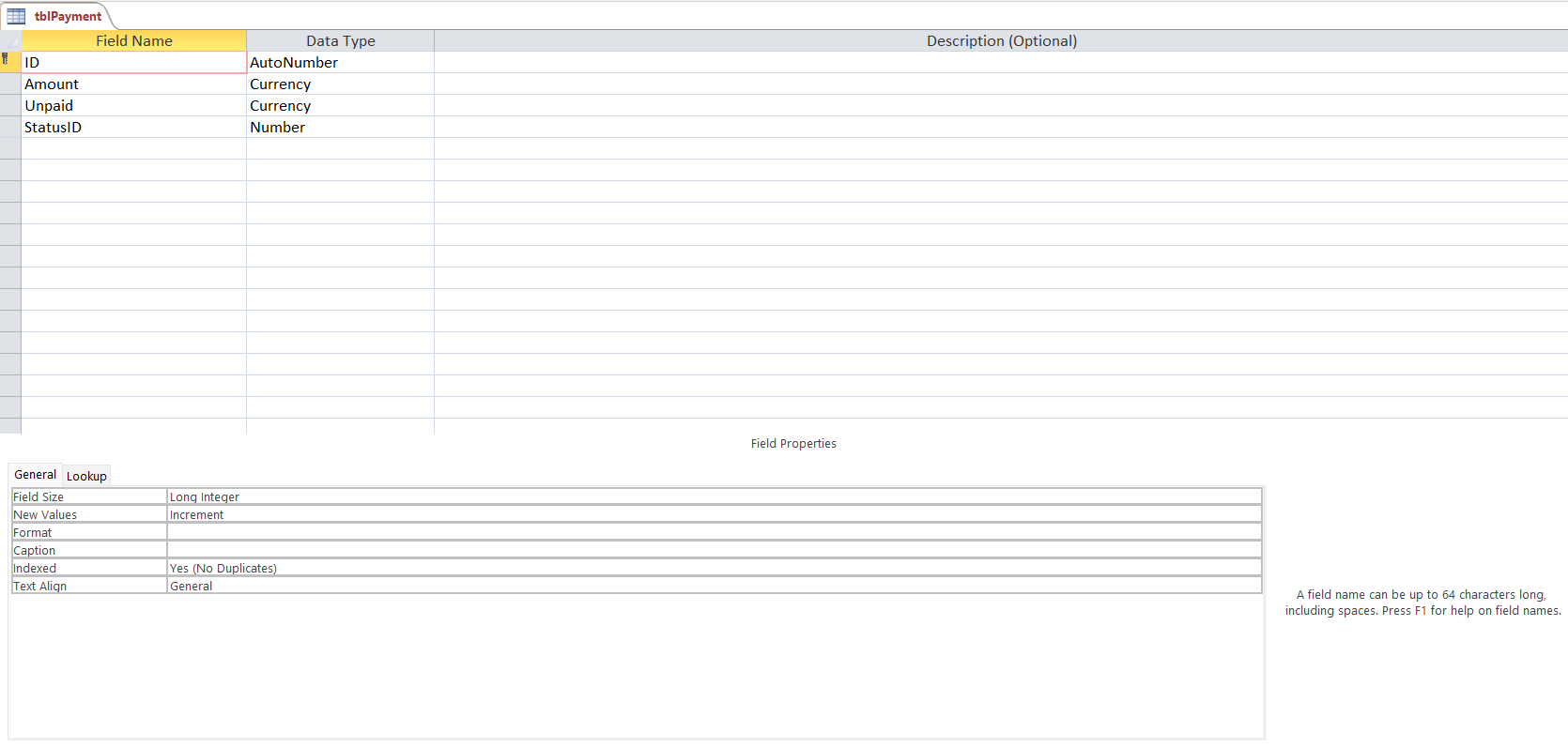
* ID : លេខសម្គាល់
* Status : ស្ថានភាពវគ្គសិក្សា



៣.៥.៩ Table Payment

នៅក្នុង Table Payment មានព័ត៌មានដូចជា

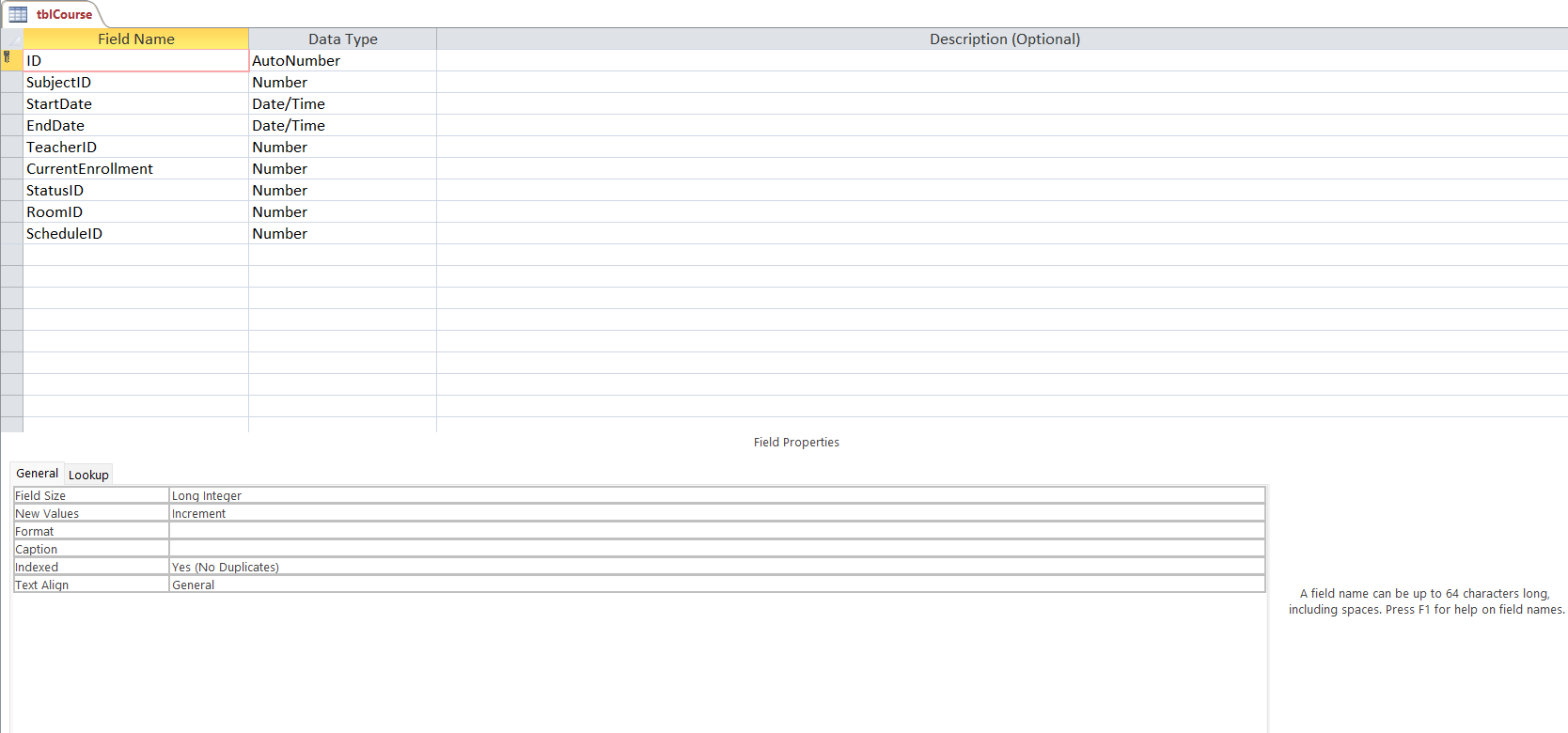
* ID : លេខសម្គាល់
* Amount : ចំនួនទឹកប្រាក់
* Unpaid : មិនទាន់បង់ប្រាក់
* StatusID : លេខសម្គាល់ស្ថានភាព



៣.៥.១០ Table Course

នៅក្នុង Table Course មានព័ត៌មានដូចជា

* ID : លេខសម្គាល់
* SubjectID : លេខសម្គាល់មុខវិជ្ជា
* StartDate : ថ្ងៃខែឆ្នាំចាប់ផ្ដើម
* EndDate : ថ្ងៃខែឆ្នាំបញ្ចប់
* TeacherID : លេខសម្គាល់គ្រូបង្រៀន
* CurrentEnrollment : ការចុះឈ្មោះបច្ចុប្បន្ន
* StatusID : លេខសម្គាល់ស្ថានភាព
* RoomID : លេខសម្គាល់បន្ទប់រៀន
* ScheduleID : លេខសម្គាល់កាលវិភាគ



ជំពូក​​ ៤

លទ្ធផលនៃការស្រាវជ្រាវ

៤.១ ការគ្រោងបង្កើត User Interface

៤.១.១ Login Form

Login Form គឺជា Form សម្រាប់អោយអ្នកប្រើប្រាស់ (User) ចុះឈ្មោះចូលប្រើប្រាស់កម្មវិធី​។

នៅក្នុង​ Login Form មានព័ត៌មានដូចជា

* UserName : សម្រាប់បញ្ចូលឈ្មោះអ្នកប្រើប្រាស់
* Password : សម្រាប់បញ្ចូលលេខសម្ងាត់
* រូបភ្នែក : សម្រាប់បិទនិងបង្ហាញលេខសម្ងាត់
* ចូលប្រើប្រាស់ : ចុចប៊ូតុងនេះ បន្ទាប់ពីបញ្ចូលឈ្មោះនិងលេខសម្ងាត់ដែលត្រឹមត្រូវរួច
* ចាកចេញ : ចុចប៊ូតុងនេះ ប្រសិនបើអ្នកចង់ចាកចេញពីកម្មវិធី



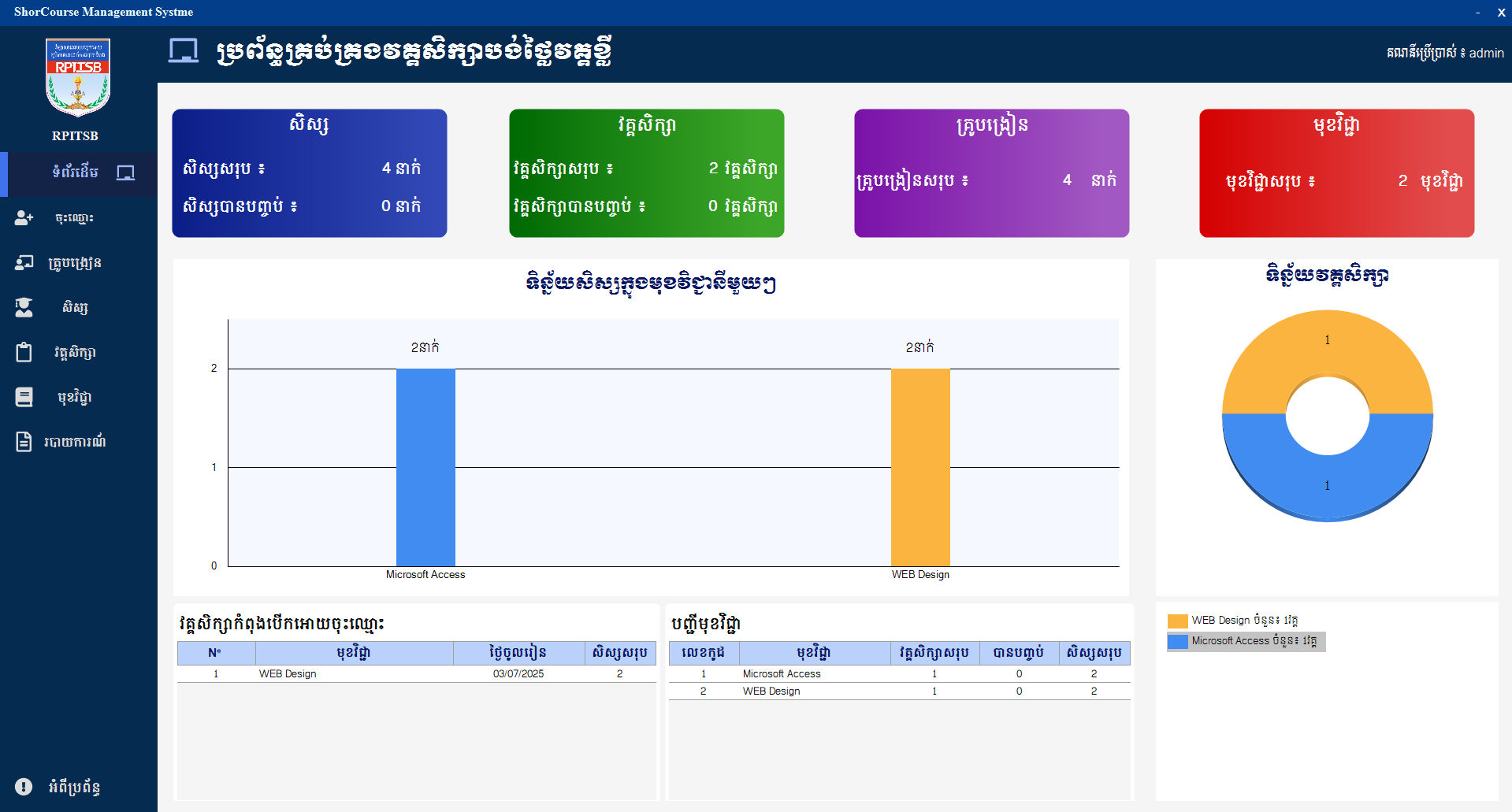
រូបភាពទី ៤.១.១ Login Form

៤.១.២ Interface Front End

Interface Front End គឺជាផ្ទាំងសម្រាប់ស្វាគមន៍ការចូលមកប្រើប្រាស់កម្មវិធី​ និងជាផ្ទាំងចម្បងដែលគ្រប់គ្រងទៅលើ Form ទាំងអស់នៅក្នុងកម្មវិធី លើកលែងតែ Login Form ។

នៅក្នុង Interface Front End​ មានធាតុសំខាន់ៗដូចជា

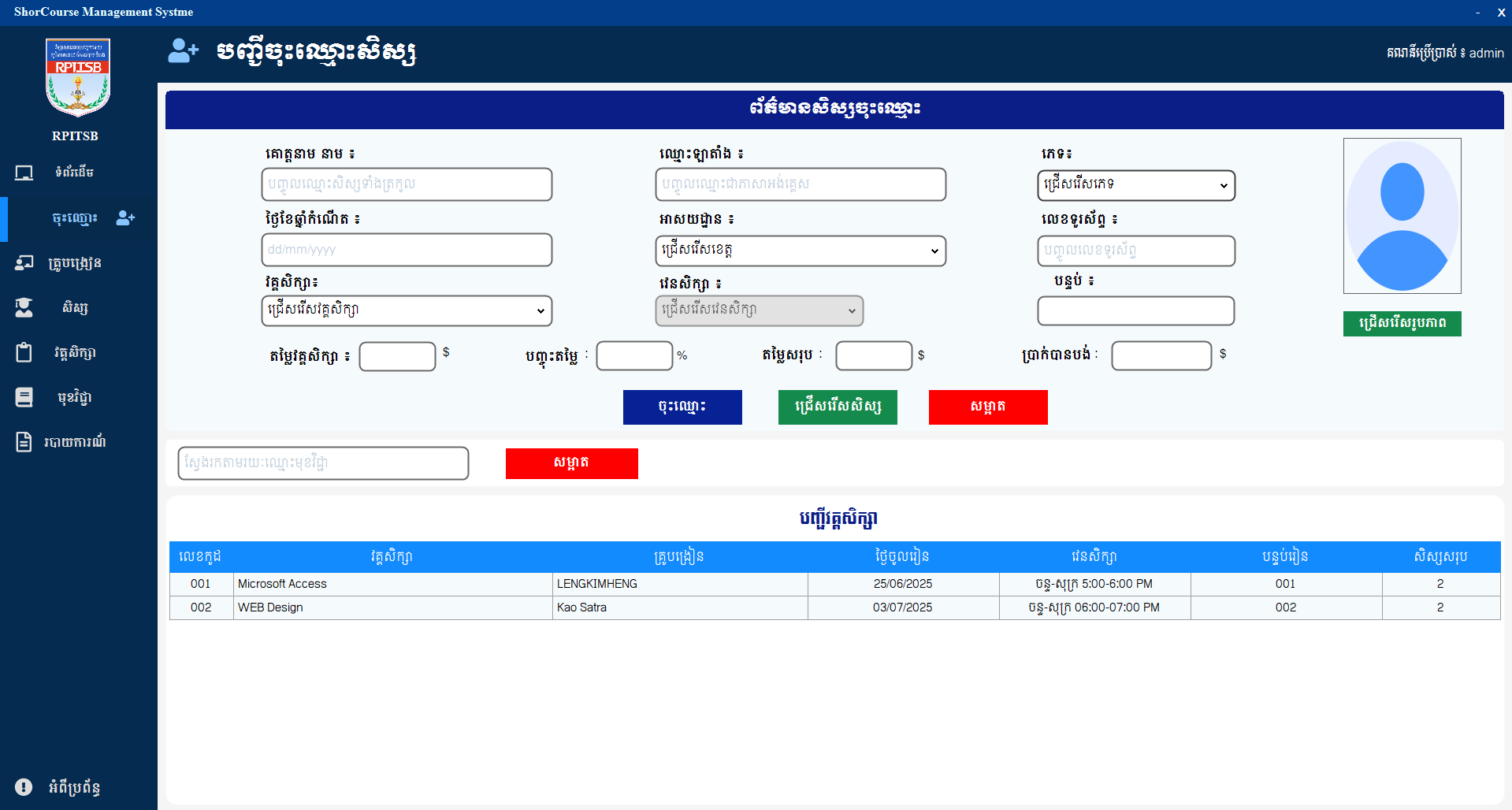
* បញ្ជីចុះឈ្មោះសិស្ស : ប៊ូតុងនេះសម្រាប់ចូលទៅកាន់ បញ្ជីចុះឈ្មោះសិស្ស
* បញ្ជីព័ត៌មានគ្រូបង្រៀន : ប៊ូតុងនេះសម្រាប់ចូលទៅកាន់ បញ្ជីព័ត៌មានគ្រូបង្រៀន
* បញ្ជីព័ត៌មានសិស្ស : ប៊ូតុងនេះសម្រាប់ចូលទៅកាន់ បញ្ជីព័ត៌មានសិស្ស
* បញ្ជីព័ត៌មានវគ្គសិក្សា : ប៊ូតុងនេះសម្រាប់ចូលទៅកាន់ បញ្ជីព័ត៌មានវគ្គសិក្សា
* បញ្ជីព័ត៌មានមុខវិជ្ជា : ប៊ូតុងនេះសម្រាប់ចូលទៅកាន់ បញ្ជីព័ត៌មានមុខវិជ្ជា
* បញ្ជីរបាយការណ៍ : ប៊ូតុងនេះសម្រាប់ចូលទៅកាន់ បញ្ជីរបាយការណ៍



រូបភាពទី ៤.១.២ Interface Front End

៤.១.២.១ បញ្ជីចុះឈ្មោះសិស្ស

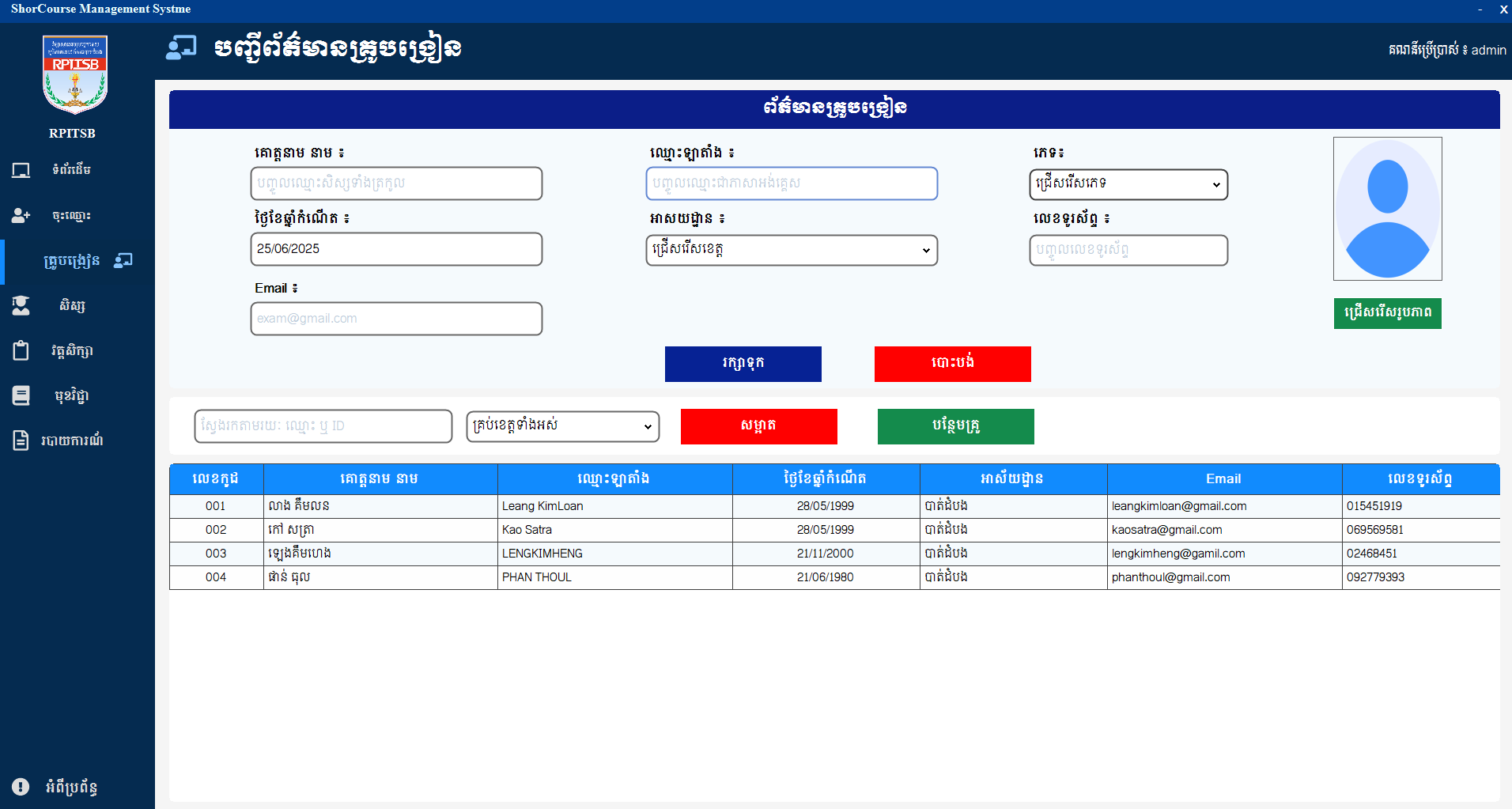
នៅក្នុងបញ្ជីចុះឈ្មោះសិស្ស គឺជាបញ្ជីសម្រាប់បញ្ចូលព័ត៌មានសិស្ស មានព័ត៌មានតាមការណែនាំដូចនៅក្នុង Form ខាងក្រោម ដូចជា ព័ត៌មានសិស្សចុះឈ្មោះ បញ្ជីវគ្គសិក្សា ។



រូបភាពទី ៤.១.២.១ បញ្ជីចុះឈ្មោះសិស្ស

៤.១.២.២ បញ្ជីព័ត៌មានគ្រូបង្រៀន

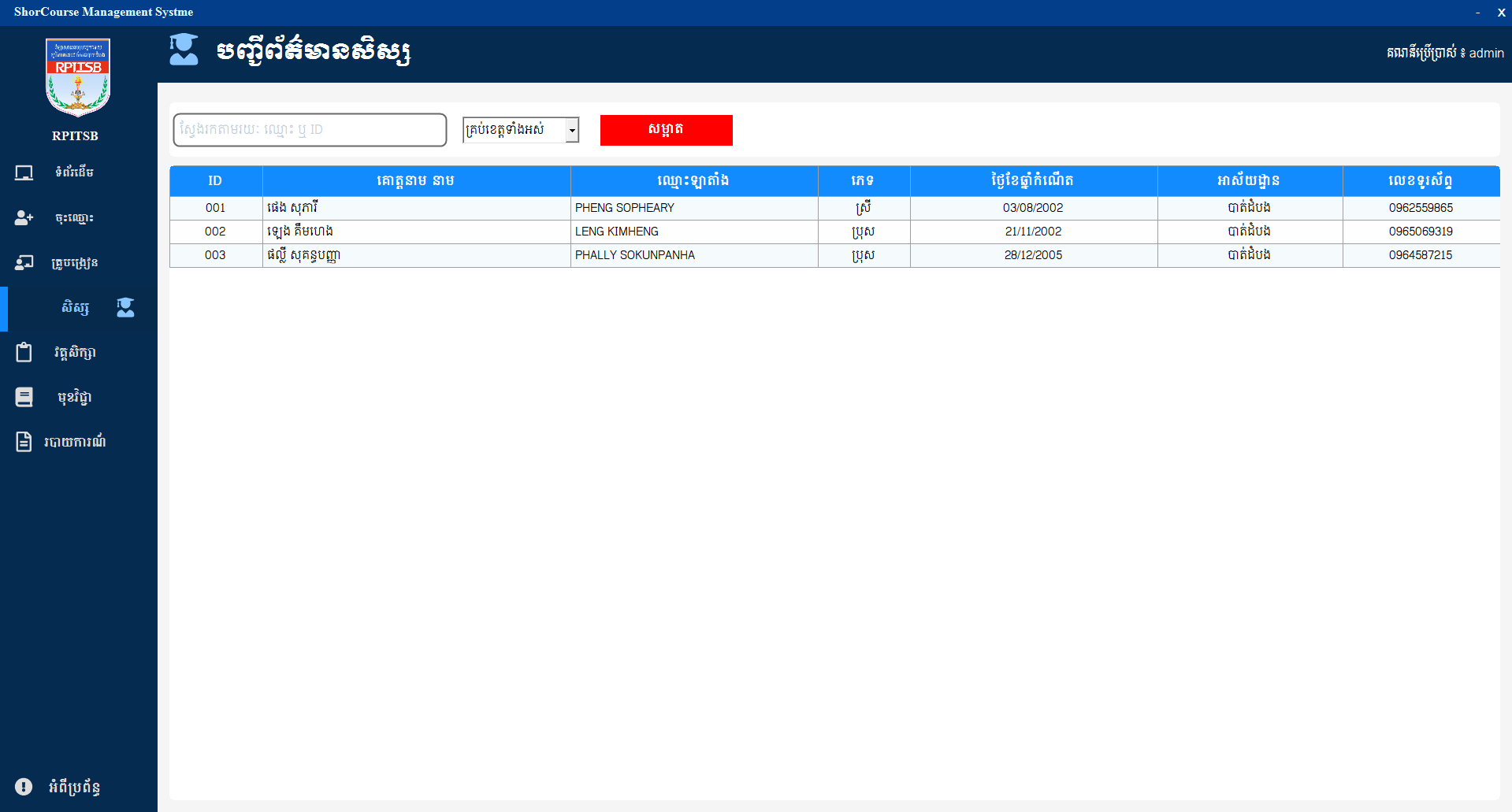
នៅក្នុងបញ្ជីព័ត៌មានគ្រូបង្រៀន ​គឺជាបញ្ជីសម្រាប់បញ្ចូលព័ត៌មានគ្រូបង្រៀន ដែលមានបង្ហាញឲ្យដឹងពីព័ត៌មានរបស់គ្រូបង្រៀន និងមធ្យោបាយសម្រាប់ធ្វើការទំនាក់ទំនងទៅកាន់ពួកគាត់។



រូបភាពទី ៤.១.២.២ បញ្ជីព័ត៌មានគ្រូបង្រៀន

៤.១.២.៣ បញ្ជីព័ត៌មានសិស្ស

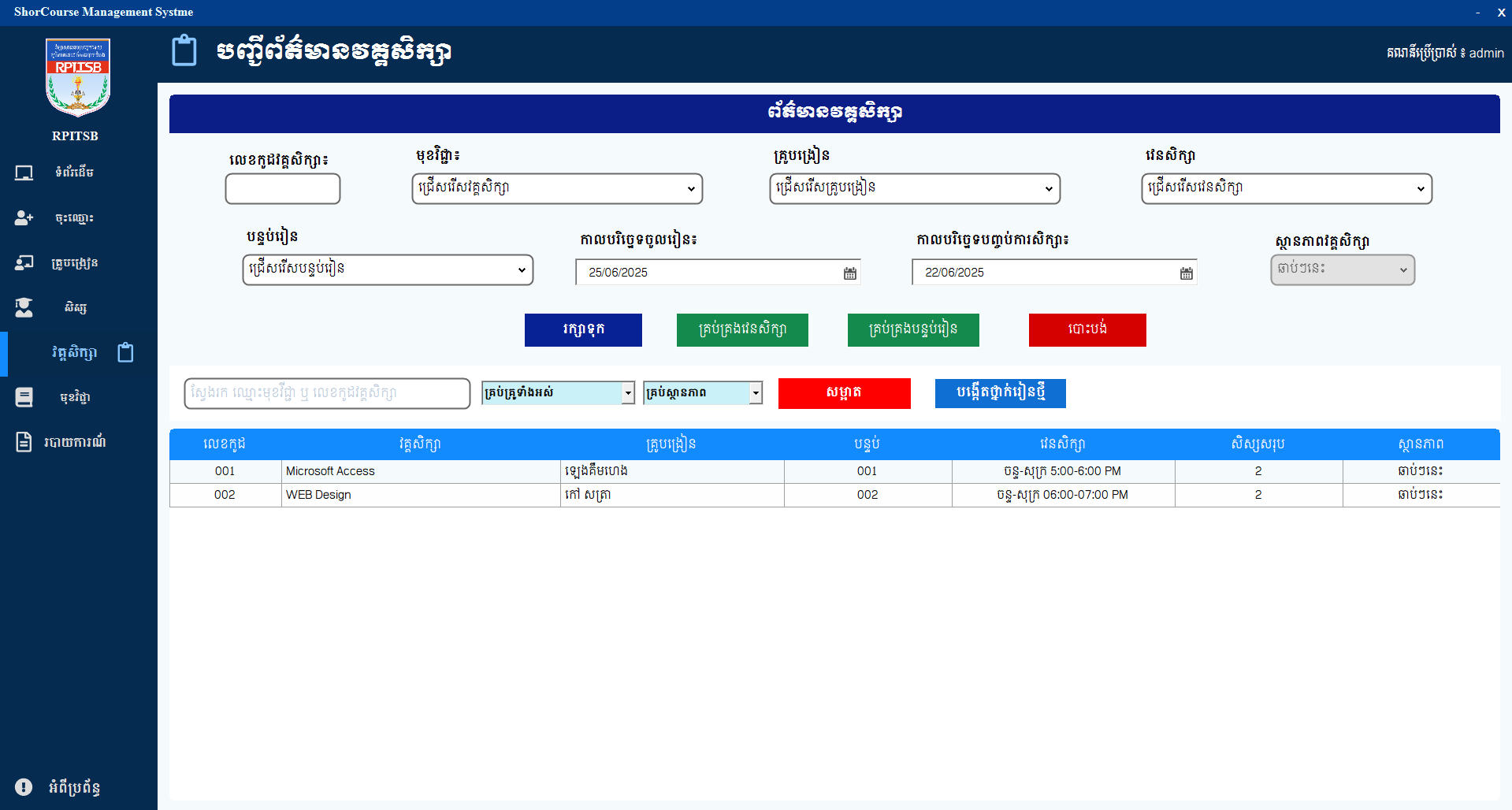
នៅក្នុងបញ្ជីព័ត៌មានសិស្ស គឺជាបញ្ជីសម្រាប់ផ្ទុកនិងបង្ហាញព័ត៌មានរបស់សិស្សម្នាក់ៗ ដែលបានចុះឈ្មោះចូលរៀន។



រូបភាពទី ៤.១.២.៣ បញ្ជីព័ត៌មានសិស្ស

៤.១.២.៤ បញ្ជីព័ត៌មានវគ្គសិក្សា

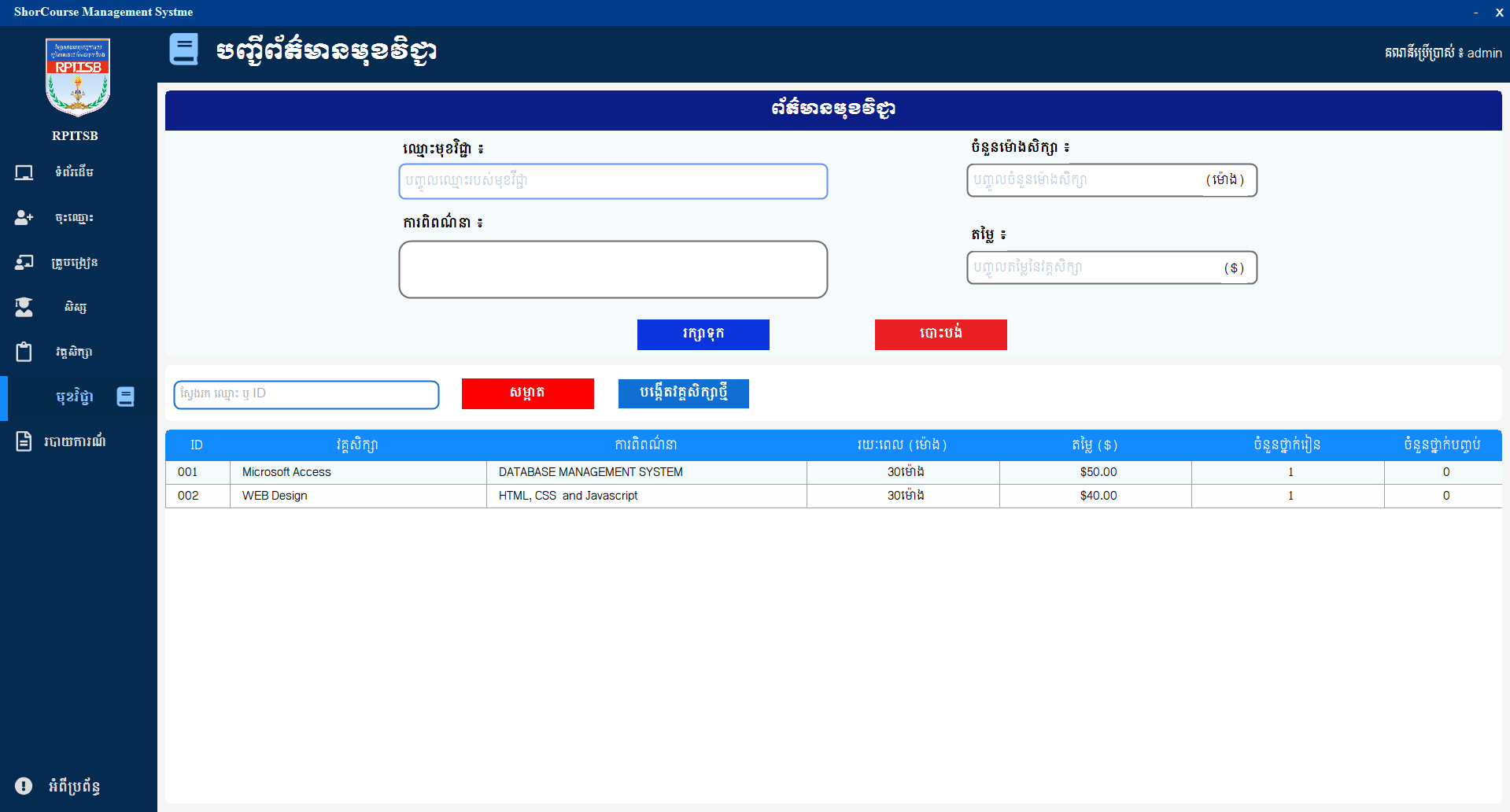
នៅក្នុងបញ្ជីព័ត៌មានវគ្គសិក្សា គឺជាបញ្ជីសម្រាប់បង្កើតថ្នាក់រៀន ហើយបង្ហាញពីវគ្គសិក្សា គ្រូត្រូវបង្រៀន បន្ទប់រៀន វេនសិក្សា សិស្សសរុបដែលចុះឈ្មោះរៀនក្នុងវគ្គសិក្សានោះ និងស្ថានភាពត្រូវសិក្សា។



រូបភាពទី ៤.១.២.៤ បញ្ជីព័ត៌មានវគ្គសិក្សា

៤.១.២.៥ បញ្ជីព័ត៌មានមុខវិជ្ជា

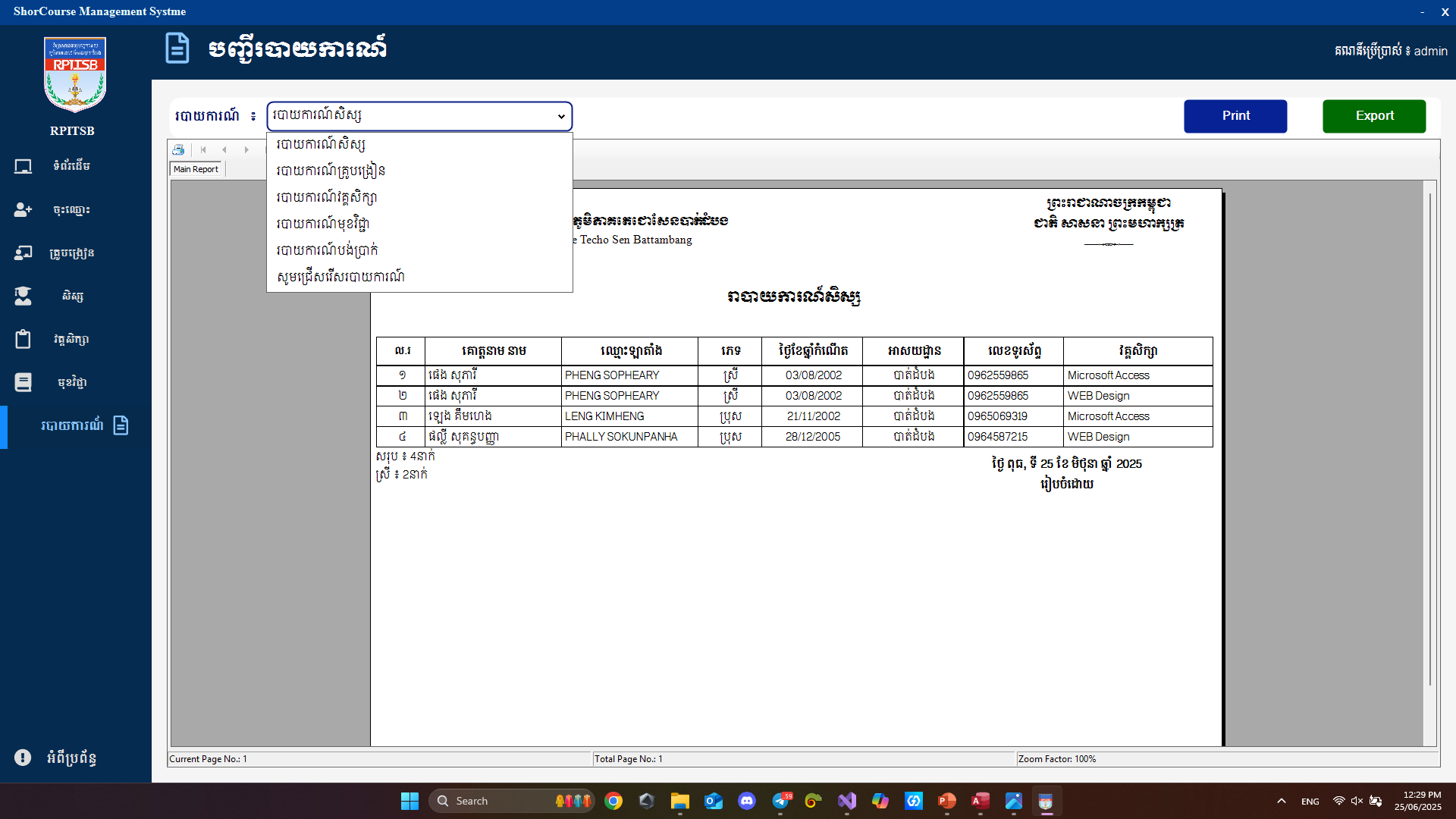
នៅក្នុងបញ្ជីព័ត៌មានមុខវិជ្ជា គឺមានទម្រង់ព័ត៌មានមុ​ខវិជ្ជា សម្រាប់បង្ហាញឲ្យសិស្សានុសិស្សដឹងអំពី មុខវិជ្ជាដែលសាលាមាន តម្លៃតាមមុខវិជ្ជានីមួយៗ ម៉ោងដែលត្រូវសិក្សា ដើម្បីអោយសិស្សជ្រើសរើសតាមបំណងនៃការសិក្សា។

****

រូបភាពទី ៤.១.២.៥ បញ្ជីព័ត៌មានមុខវិជ្ជា

៤.១.២.៥ បញ្ជីរបាយការណ៍

នៅក្នុងបញ្ជីរបាយការណ៍​មាន​ព័ត៌មានទាក់ទងទៅនឹង ចំនួនគ្រូបង្រៀន ចំនួនសិស្ស សិស្សស្រី​ សិស្សប្រុស វគ្គសិក្សា​ មុខវិជ្ជា ការបង់ប្រាក់។



រូបភាពទី ៤.១.២.៦ បញ្ជីរបាយការណ៍

ជំពូក​​ ៥

សន្និដ្ឋាន និង ការផ្តល់អនុសាសន៏

៥.១ សេចក្តីសន្និដ្ឋាន

តាមរយៈការសិក្សាខាងលើ ប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី) ដែលធ្វើការដោយប្រើប្រាស់កម្មវិធី Microsoft Office (Excel) ដូចនេះក្រុមអ្នកសិក្សាបានសង្កេតឃើញថា ប្រព័ន្ធគ្រប់គ្រងនាពេលបច្ចុប្បន្នគឺមានភាពយឺតយ៉ាវ ស្មុគស្មាញ ប្រើពេលវេលាយូរក្នុងការវាយបញ្ចូល និងមិនតែប៉ុណ្ណោះលក្ខណៈគ្រប់គ្រងទិន្នន័យពុំបានច្បាស់លាស់ទៀតផង ព្រោះថាមិនទាន់មានការប្រើប្រាស់តាមបច្ចេកវិទ្យាព័ត៌មានទំនើបនៅឡើយ ហើយនិងរាល់ការងារទាំងមូលនោះតែងតែជួបបញ្ហាជាញឹកញាប់ ។

ដោយយោងតាមមូលហេតុ និងចំណុចចខ្វះខាតនានា ដែលប្រព័ន្ធការងារចាស់ធ្លាប់ជួបប្រទះជាហេតុនាំឱ្យត្រូវការកែសម្រួលប្រព័ន្ធនេះឡើងវិញ។ ដូច្នេះហើយទើបក្រុមអ្នកសិក្សាបានធ្វើការស្នើសុំដើម្បីសាងសង់ប្រព័ន្ធគ្រប់គ្រងតាមបែបថ្មី ដោយបច្ចេកវិទ្យាព័ត៌មានទំនើបមួយក្នុងគោលបំណងកែប្រែប្រព័ន្ធចាស់ដែលស្មុគស្មាញនោះ ដោយប្រព័ន្ធថ្មីនេះធ្វើការកែប្រែការងារដែលស្មុគស្មាញទៅជាប្រព័ន្ធដែលមានលក្ខណៈសាមញ្ញ ងាយយល់ និងងាយស្រួលប្រើ ថែមទាំងមានប្រសិទ្ធភាពខ្ពស់សម្រាប់អ្នកប្រើប្រាស់ ហើយប្រព័ន្ធថ្មីនេះដែរមានឈ្មោះថា ប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី) ។ ក្នុងការបង្កើតប្រព័ន្ធនេះក្រុមនិស្សិតយើងខ្ញុំបានសម្រេចចិត្តប្រើប្រាស់កម្មវិធី Visual Studio និង Microsoft Access សម្រាប់ធ្វើជា Database គឺជាកន្លែងផ្ទុកទិន្នន័យ និងកម្មវិធី Crystal Report ដើម្បីបង្កើតជារបាយការណ៍ (Report) ផ្សេងៗ ។

ក្រោយពីការអនុវត្តន៍នូវការងាររបស់ប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី) តាមបែបថ្មីនេះរួចមកធ្វើឱ្យយើងខ្ញុំសង្កេតឃើញថាវាមានលក្ខណៈល្អប្រសើរជាង ប្រព័ន្ធគ្រប់គ្រងចាស់ ព្រោះថាក្រោយពីពេលដែលក្រុមនិស្សិតយើងខ្ញុំបានអនុវត្តន៍ប្រព័ន្ធនេះរួចមកឃើញថាការគ្រប់គ្រងនូវព័ត៌មានសិស្សមានលក្ខណៈប្រសើរមួយចំនួនដូចខាងក្រោម៖

* ប្រព័ន្ធថ្មីនេះមានសុវត្ថិភាពខ្ពស់
* អាចធ្វើការស្វែងរកព័ត៌មានសិស្ស បានលឿន
* អាចកាត់បន្ថយភាពស្ទួន
* បង្កើនល្បឿនការងារ និងពុំមានភាពស្មុគស្មាញ
* ជាប្រព័ន្ធគ្រប់គ្រងមួយដែលបានវិភាគ និងរៀបចំបានច្បាស់លាស់
* អាចរក្សាទិន្នន័យទុកបានច្រើន និងភាពងាយស្រួលគ្រប់គ្រងទិន្នន័យ
* ផ្តល់ភាពងាយស្រួលដល់អ្នកប្រើប្រាស់ក្នុងការបញ្ចូលទិន្នន័យ បង្ហាញទិន្នន័យ និងបោះពុម្ព
* មានលក្ខណៈងាយស្រួលក្នុងការ Update Data ហើយមិនងាយបាត់បង់
* ទទួលបាននូវប្រព័ន្ធថ្មីមួយមានលក្ខណៈទំនើប និងងាយស្រួលប្រើប្រាស់
* ទទួលបាននូវចំណេះដឹងដ៏ធំដល់ក្រុមនិស្សិតយើងខ្ញុំ

សរុបសេចក្តីមកប្រព័ន្ធគ្រប់គ្រងវគ្គសិក្សាបង់ថ្លៃ(វគ្គខ្លី) ដែលក្រុមនិស្សិតយើងខ្ញុំបានកសាងថ្មីនេះពិតជាមានសារៈសំខាន់យ៉ាងខ្លាំងក្នុងការជួយសម្រួលដល់ការងាររបស់វិទ្យាស្ថាន ពីព្រោះវាមានសុវត្ថិភាពក្នុងការរក្សាទិន្នន័យ ងាយស្រួលបញ្ចូល លុប និងកែប្រែទិន្នន័យ មិនតែប៉ុណ្ណោះវាមានភាពងាយស្រួលក្នុងការស្វែងរកទិន្នន័យដែលទាក់ទងទៅនឹងព័ត៌មានសិស្ស​ ឬព័ត៌មានគ្រូ បានយ៉ាងលឿន រហ័សទាន់ចិត្ត ចំណែកការផ្ទុកទិន្នន័យវិញក៏បានច្រើនដែរ ។

៥.២ ការផ្តល់អនុសាសន៏

បន្ទាប់ពីការអនុវត្តនូវកម្មវីធី​ ​ ប្រព័ន្ធគ្រប់គ្រង​វគ្គសិក្សាបង់ថ្លៃវគ្គខ្លី មកក្រុមនិស្សិតយើងខ្ញុំអាចសន្និដ្ឋានបានថាប្រព័ន្ធនេះអាចយកទៅធ្វើការប្រើប្រាស់សម្រាប់គ្រប់គ្រង ហើយមានលក្ខណៈងាយស្រួលប្រើជាងប្រព័ន្ធចាស់ ព្រោះវាមានដំណើរការលឿនពុំមានលក្ខណៈសាំញ៉ាំសម្រាប់ការបញ្ជូលទិន្នន័យឡើយ ។ ទោះជាយ៉ាងក៏ដោយប្រព័ន្ធនេះមានរយៈពេលខ្លី ។

**ឯកសារយោង**

**ឯកសារភាសាខ្មែរ**

* **អេង សុវណ្ណឫទ្ធិ (២០១០) ការអភិវឌ្ឍប្រព័ន្ធគ្រប់គ្រងលិខិតរដ្ឋបាល ដោយប្រើប្រាស់កម្មវិធី Visual** Studio.Net 2005និង Microsoft Access 2003

**ឯកសារភាសាអង់គ្លេស**

* Fred R. McFadden, Jeffrey A, Hoffer, Mary B.(?)n Prescott Modern database Management
* Gary B. Shelly, Thomas J. Rosenblatt(?)System Analysis and Design 4th Edition
* Kenneth C.Laudon, Jane P.Laudon(?)Management Information System 7 th Edition
* Jujitsu Learning Media Limited (2000) Database System Design
* Jujitsu Learning Media Limited (2000) Database System Designer

**ឯកសារពី Internet**

* Channel Youtube: DC SoftwareSolution
* Website: Wikipedia the Free Encyclopedia(https://en.wikipedia.org/)
* Website: Koun Kodee (http://kounkoder.blogspot.com/2012/03/software.html)
* Website: Stack Overflow (http://stacjoverflow.com)

**ឧបសម្ព័ន្ធ**

ឧបសម្ព័ន្ធ ក-លិខិតស្នើសុំចុះសិក្សាស្រាវជ្រាវ

ឧបសម្ព័ន្ធ ខ-លិខិតស្នើសុំចុះសិក្សាស្រាវជ្រាវ

ឧបសម្ព័ន្ធ គ-បញ្ចីឈ្មោះនិស្សិតចុះស្រាវជ្រាវបញ្ចប់ការសិក្សាថ្នាក់ពត៌មានវិទ្យា

ប្រធានបទ “ប្រព័ន្ធគ្រប់គ្រងបុគ្គលិក RPITSB”

ឧបសម្ព័ន្ធ ឃ -សកម្មភាពជួបជាមួយគ្រូដឹកនាំ

ឧបសម្ព័ន្ធ ង -សកម្មភាពជួបសម្ភាសអនុប្រធានការិយាល័យរដ្ឋបាល

Source Code

**1.LoginForm**

Public Class LoginForm

Public Sub New()

' This call is required by the designer.

InitializeComponent()

' Add any initialization after the InitializeComponent() call.

Me.FormBorderStyle = FormBorderStyle.None

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles btnExit.Click

Me.Close()

End Sub

Private Sub Button1\_Click\_1(sender As Object, e As EventArgs) Handles btnLogin.Click

If CheckField() Then

If User.CheckLogin(txtUserName.Text.Trim(), txtPassWord.Text.Trim()) Then

MainForm.Visible = True

Me.Hide()

Else

MessageBox.Show("ឈ្មោះអ្នកប្រើប្រាស់ ឬ ពាក្យសម្ងាត់មិនត្រឹមត្រូវទេ", "ការព្រមាន", MessageBoxButtons.OK, MessageBoxIcon.Warning)

txtUserName.Focus()

End If

End If

End Sub

Function CheckField() As Boolean

If txtUserName.Text = "" Then

MessageBox.Show("សូមបញ្ចូលឈ្មោះអ្នកប្រើប្រាស់", "ការព្រមាន", MessageBoxButtons.OK, MessageBoxIcon.Warning)

txtUserName.Focus()

Return False

Else

If txtPassWord.Text = "" Then

MessageBox.Show("សូមបញ្ចូលពាក្យសម្ងាត់", "ការព្រមាន", MessageBoxButtons.OK, MessageBoxIcon.Warning)

txtPassWord.Focus()

Return False

Else

Return True

End If

End If

End Function

Private Sub btnShowHide\_Click(sender As Object, e As EventArgs) Handles btnShowHide.Click

If txtPassWord.UseSystemPasswordChar = True Then

txtPassWord.UseSystemPasswordChar = False

btnShowHide.IconChar = FontAwesome.Sharp.IconChar.EyeSlash

Else

txtPassWord.UseSystemPasswordChar = True

btnShowHide.IconChar = FontAwesome.Sharp.IconChar.Eye

End If

End Sub

Private Sub btnClose\_Click(sender As Object, e As EventArgs) Handles btnClose.Click

Me.Close()

End Sub

Private Sub txtPassWord\_KeyDown(sender As Object, e As KeyEventArgs) Handles txtPassWord.KeyDown

If e.KeyCode = Keys.Enter Then

btnLogin.PerformClick()

End If

End Sub

End Class

**2.MainForm**

Imports FontAwesome.Sharp

Public Class MainForm

Private currentBtn As IconButton

Private leftBorderBtn As Panel

Private currentChildForm As Form

Public login As Integer

'Constructor

Public Sub New**()**

' This call is required by the designer.

InitializeComponent**()**

'Add any initialization after the InitializeComponent() call.

Me.FormBorderStyle **=** FormBorderStyle.None

leftBorderBtn **=** New Panel**()**

leftBorderBtn.Size **=** New Size**(10,** btnReport.Height**)**

panelMenu.Controls.Add**(**leftBorderBtn**)**

Me.DoubleBuffered **=** True

Me.MaximizedBounds **=** Screen.FromHandle**(**Me.Handle**).**WorkingArea

Dim test As LoginForm **=** New LoginForm**()**

login **=** test.ShowDialog**()**

lbUsername.Text **=** lbUsername.Tag.ToString**()** **&** User.userName

End Sub

Private Sub Activatebutton**(**senderBtn As Object**,** customColor As Color**)**

If senderBtn IsNot Nothing Then

DisableButton**()**

currentBtn **=** CType**(**senderBtn**,** IconButton**)**

currentBtn.BackColor **=** ThemeColor.ChangeColorBrightness**(**customColor**,** **-0.7)**

currentBtn.ForeColor **=** ThemeColor.ChangeColorBrightness**(**customColor**,** **0.5)**

currentBtn.IconColor **=** ThemeColor.ChangeColorBrightness**(**customColor**,** **0.5)**

currentBtn.Font **=** New Font**(**currentBtn.Font.FontFamily**,** **12.0,** FontStyle.Bold**)**

If panelMenu.Width **>=** **200** Then

currentBtn.ImageAlign **=** ContentAlignment.MiddleRight

currentBtn.Padding **=** New Padding**(10,** **0,** **20,** **0)**

End If

'Left Border

leftBorderBtn.BackColor **=** customColor

leftBorderBtn.Location **=** New Point**(0,** currentBtn.Location.Y**)**

leftBorderBtn.Visible **=** True

leftBorderBtn.BringToFront**()**

'Current Form Icon

IconCurrentForm.IconChar **=** currentBtn.IconChar

IconCurrentForm.IconColor **=** ThemeColor.ChangeColorBrightness**(**customColor**,** **0.5)**

'Current Form Title

lbFormTitle.Text **=** LTrim**(**currentBtn.Tag.ToString**)**

Dim middle As Integer **=** **((**panelTitlebar.Size.Width **/** **2)** **-** **(**lbFormTitle.Size.Width **/** **2))**

lbFormTitle.Location **=** New Point**(**middle**,** lbFormTitle.Location.Y**)**

End If

End Sub

Private Sub DisableButton**()**

If currentBtn IsNot Nothing Then

currentBtn.BackColor **=** ThemeColor.menuBackColor

currentBtn.ForeColor **=** Color.Gainsboro

currentBtn.IconColor **=** Color.Gainsboro

currentBtn.Font **=** New Font**(**currentBtn.Font.FontFamily**,** **10.0,** FontStyle.Bold**)**

currentBtn.ImageAlign **=** ContentAlignment.MiddleLeft

currentBtn.Padding **=** New Padding**(10,** **0,** **20,** **0)**

If panelMenu.Width **<** **200** Then

currentBtn.ImageAlign **=** ContentAlignment.MiddleCenter

End If

End If

End Sub

'Open Child Form

Private Sub OpenChildForm**(**childForm As Form**)**

If currentChildForm IsNot Nothing Then

currentChildForm.Close**()**

End If

currentChildForm **=** childForm

childForm.TopLevel **=** False

childForm.FormBorderStyle **=** FormBorderStyle.None

childForm.Dock **=** DockStyle.Fill

panelChildForm.Controls.Add**(**childForm**)**

'panelChildForm.Tag = childForm

childForm.BringToFront**()**

childForm.Show**()**

End Sub

Private Sub Reset**()**

DisableButton**()**

leftBorderBtn.Visible **=** False

IconCurrentForm.IconChar **=** IconChar.Home

IconCurrentForm.IconColor **=** Color.Gainsboro

lbFormTitle.Text **=** "Home"

End Sub

Private Sub BtnDashboard\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnDashboard.Click

Activatebutton**(**sender**,** ThemeColor.DashboardColor**)**

OpenChildForm**(**New DashboardForm**())**

End Sub

Private Sub BtnStudent\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnRegister.Click

Activatebutton**(**sender**,** ThemeColor.activeColor**)**

OpenChildForm**(**New RegisterForm**())**

End Sub

Private Sub btnClose\_Click**(**sender As Object**,** e As EventArgs**)**

Application.Exit**()**

End Sub

Private Sub btnCourse\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnCourse.Click

Activatebutton**(**sender**,** ThemeColor.activeColor**)**

OpenChildForm**(**New SubjectForm**())**

End Sub

Private Sub btnTeacher\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnTeacher.Click

Activatebutton**(**sender**,** ThemeColor.activeColor**)**

OpenChildForm**(**New TeachersForm**())**

End Sub

Private Sub btnStudent\_Click\_1**(**sender As Object**,** e As EventArgs**)** Handles btnStudent.Click

Activatebutton**(**sender**,** ThemeColor.activeColor**)**

OpenChildForm**(**New StudentForm**())**

End Sub

Private Sub btnClass\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnClass.Click

Activatebutton**(**sender**,** ThemeColor.activeColor**)**

OpenChildForm**(**New CourseForm**())**

End Sub

Private Sub btnClose\_Click\_1**(**sender As Object**,** e As EventArgs**)** Handles btnClose.Click

Application.Exit**()**

End Sub

Private Sub btnMin\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnMin.Click

Me.WindowState **=** FormWindowState.Minimized

End Sub

Private Sub MainForm\_Load**(**sender As Object**,** e As EventArgs**)** Handles MyBase.Load

If login **=** **2** Then

Application.Exit**()**

End If

btnDashboard.PerformClick**()**

End Sub

Private Sub btnReport\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnReport.Click

Activatebutton**(**sender**,** ThemeColor.activeColor**)**

OpenChildForm**(**New ReportForm**())**

End Sub

Private Sub btnAbout\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnAbout.Click

Activatebutton**(**sender**,** ThemeColor.activeColor**)**

OpenChildForm**(**New AboutSystemForm**())**

End Sub

End Class

**3.DashboardForm**

Imports System.Windows.Forms.DataVisualization.Charting

Imports Guna.UI2.WinForms

Public Class DashboardForm

Dim dashboard As New Dashboard()

Private Sub DashboardForm\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

lbTotalStudnet.Text = dashboard.totalStudent.ToString()

lbCompleteStudent.Text = dashboard.completeStudent.ToString()

lbTotalClass.Text = dashboard.totalCourse

lbCompleteClass.Text = dashboard.completeCourse

lbTotalTeacher.Text = dashboard.totalTeacher

lbTotalCourse.Text = dashboard.TotalSubject

'Insert Data to chart

StudentChart.DataSource = dashboard.CourseList

StudentChart.Series(0).XValueMember = "Key"

StudentChart.Series(0).YValueMembers = "Value"

StudentChart.DataBind()

StudentChart.Legends(0).CustomItems.Clear()

CourseChart.DataSource = dashboard.SubjectList

CourseChart.Series(0).XValueMember = "Key"

CourseChart.Series(0).YValueMembers = "Value"

CourseChart.DataBind()

'Insert Data To DG CourseCoomingSoon

CoomingCouseList.DataSource = dashboard.CourseCoomingSoon()

With CoomingCouseList

.Columns(0).Width = 60

.Columns(0).HeaderText = "Nº"

.Columns(0).SortMode = DataGridViewColumnSortMode.NotSortable

.Columns(0).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter

.Columns(1).Width = 150

.Columns(1).HeaderText = "មុខវិជ្ជា"

.Columns(1).SortMode = DataGridViewColumnSortMode.NotSortable

.Columns(1).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleLeft

.Columns(2).Width = 100

.Columns(2).HeaderText = "ថ្ងៃចូលរៀន"

.Columns(2).SortMode = DataGridViewColumnSortMode.NotSortable

.Columns(2).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter

.Columns(3).Width = 90

.Columns(3).HeaderText = "សិស្សសរុប"

.Columns(3).SortMode = DataGridViewColumnSortMode.NotSortable

.Columns(3).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter

End With 'Insert Data To DG CourseCoomingSoon

dgSubjectList.DataSource = dashboard.GetSubjectList()

With dgSubjectList

.Columns(0).Width = 80

.Columns(0).HeaderText = "លេខកូដ"

.Columns(0).SortMode = DataGridViewColumnSortMode.NotSortable

.Columns(0).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter

.Columns(1).Width = 170

.Columns(1).HeaderText = "មុខវិជ្ជា"

.Columns(1).SortMode = DataGridViewColumnSortMode.NotSortable

.Columns(1).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleLeft

.Columns(2).Width = 100

.Columns(2).HeaderText = "វគ្គសិក្សាសរុប"

.Columns(2).SortMode = DataGridViewColumnSortMode.NotSortable

.Columns(2).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter

.Columns(3).Width = 90

.Columns(3).HeaderText = "បានបញ្ចប់"

.Columns(3).SortMode = DataGridViewColumnSortMode.NotSortable

.Columns(3).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter

.Columns(4).Width = 90

.Columns(4).HeaderText = "សិស្សសរុប"

.Columns(4).SortMode = DataGridViewColumnSortMode.NotSortable

.Columns(4).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter

End With

End Sub

End Class

**4.RegisterForm**Imports System.Data.OleDb

Imports System.Data.SqlTypes

Imports System.IO

Imports CrystalDecisions.CrystalReports.Engine

Imports CrystalDecisions.Shared

Imports Guna.UI2.WinForms

Imports Microsoft.SqlServer

Public Class RegisterForm

Dim register As New Register**()**

Dim startup As Boolean **=** True

Public Sub New**()**

' This call is required by the designer.

InitializeComponent**()**

' Add any initialization after the InitializeComponent() call.

GetCbSubject**()**

GetCbAddress**()**

picStudent.ImageLocation **=** Application.StartupPath **&** "\Images\default.png"

End Sub

Sub Display**()**

DataGridView1.DataSource **=** register.GetRegisterData**()**

Regonize**()**

End Sub

Private Sub btnChooseImage\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnChooseImage.Click

OpenPic.Filter **=** "Image Files|\*.jpg;\*.jpeg;\*.png;\*.bmp"

OpenPic.Title **=** "Select a Picture"

OpenPic.InitialDirectory **=** Environment.GetFolderPath**(**Environment.SpecialFolder.MyPictures**)**

If OpenPic.ShowDialog**()** **=** DialogResult.OK Then

Dim filePath As String **=** OpenPic.FileName

picStudent.ImageLocation **=** filePath

End If

End Sub

Private Function CheckField**()** As Boolean

If txtKhName.Text.Length **<=** **0** Then

MessageBox.Show**(**"សូមបញ្ចូលឈ្មោះសិស្ស"**,** "Warning"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Warning**)**

txtKhName.Focus**()**

Return False

End If

If txtEngName.Text.Length **<=** **0** Then

MessageBox.Show**(**"សូមបញ្ចូលឈ្មោះឡាតាំងរបស់សិស្ស"**,** "Warning"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Warning**)**

txtEngName.Focus**()**

Return False

End If

If cbAddress.SelectedIndex **=** cbAddress.Items.Count **-** **1** Then

MessageBox.Show**(**"សូមជ្រើសរើសអាស័យដ្ឋាន"**,** "Warning"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Warning**)**

cbAddress.Focus**()**

Return False

End If

If txtPhone.Text.Length **=** **0** Then

Dim result **=** MessageBox.Show**(**"តើអ្នកសិស្សមិនមានលេខទូរស័ព្ទមែនទេ"**,** "Warning"**,** MessageBoxButtons.YesNo**,** MessageBoxIcon.Warning**)**

If result **=** DialogResult.No Then

txtPhone.Focus**()**

Return False

End If

ElseIf **(**txtPhone.Text.Length **<** **9** Or txtPhone.Text.Length **>** **10)** And txtPhone.Text.Length **<>** **0** Then

MessageBox.Show**(**"សូមបញ្ចូលលេខទូរស័ព្ទដែលត្រឹមត្រូវ"**,** "Warning"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Warning**)**

txtPhone.Focus**()**

Return False

End If

If txtPay.Text.Length **<=** **0** Or Not IsNumeric**(**txtPay.Text**)** Then

MessageBox.Show**(**"សូមបញ្ចូលចំនួនទឹកប្រាក់ដែលសិស្សបង់"**,** "Warning"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Warning**)**

txtPay.Focus**()**

Return False

End If

Return True

End Function

Function CheckValidation**()** As Boolean

'Check if the student has already registered for the Course

Dim studentID As Integer **=** register.student.id

Dim courseID As Integer **=** register.course.ID

Dim query As String **=** "SELECT COUNT(\*) FROM tblRegister WHERE StudentID = @StudentID AND CourseID = @CourseID"

Dim cmd As New OleDbCommand**(**query**,** register.GetConnection**())**

cmd.Parameters.AddWithValue**(**"@StudentID"**,** studentID**)**

cmd.Parameters.AddWithValue**(**"@CourseID"**,** courseID**)**

register.OpenConnection**()**

Dim count As Integer **=** Convert.ToInt32**(**cmd.ExecuteScalar**())**

register.CloseConnection**()**

If count **>** **0** Then

MessageBox.Show**(**"សូមអភ័យទោសសិស្សឈ្មោះ " **&** register.student.khName **&** " ចានចុះឈ្មោះវគ្គសិក្សានេះរូចហើយ"**,** "Warning"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Warning**)**

Return False

End If

'Check if the Student has already Other at this Schedule

Dim scheduleQuery As String **=** "SELECT Count(\*)

FROM tblCourse INNER JOIN tblRegister ON tblCourse.ID **=** tblRegister.CourseID

WHERE **(((**tblRegister.StudentID**)=**@StudentID**)** AND **((**tblCourse.StatusID**)=**@ScheduleID**));**"

cmd **=** New OleDbCommand**(**scheduleQuery**,** register.GetConnection**())**

cmd.Parameters.AddWithValue**(**"@StudentID"**,** studentID**)**

cmd.Parameters.AddWithValue**(**"@ScheduleID"**,** register.course.scheduleID**)**

register.OpenConnection**()**

count **=** Convert.ToInt32**(**cmd.ExecuteScalar**())**

If count **>** **0** Then

MessageBox.Show**(**"សូមអភ័យទោសសិស្សឈ្មោះ " **&** register.student.khName **&** " ចានចុះឈ្មោះវេនសិក្សានេះរូចហើយ"**,** "Warning"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Warning**)**

Return False

End If

Return True

End Function

'Save Image and reture the path

Function SaveImageAndReturnPath**()** As String

If Path.GetFileName**(**picStudent.ImageLocation**)** **=** "default.png" Then

Return picStudent.ImageLocation

End If

Dim source As String **=** picStudent.ImageLocation

Dim imagesDir As String **=** Path.Combine**(**Application.StartupPath**,** "Images"**)**

Dim destination As String **=** Path.Combine**(**imagesDir**,** Path.GetFileName**(**source**))**

Try

If Not Directory.Exists**(**imagesDir**)** Then

Directory.CreateDirectory**(**imagesDir**)**

End If

FileCopy**(**source**,** destination**)**

Return destination

Catch ex As Exception

MessageBox.Show**(**"Failed to copy image: " **&** ex.Message**)**

Return ""

End Try

End Function

Sub GetCbSubject**()**

cbCourse.DataSource **=** New BindingSource**(**register.GetSubjectList**(),** Nothing**)**

cbCourse.DisplayMember **=** "Key"

cbCourse.ValueMember **=** "Value"

cbCourse.SelectedIndex **=** cbCourse.Items.Count **-** **1** ' Set to the last item

End Sub

Sub GetCbAddress**()**

Dim addressData **=** register.student.GetAddressData**()**

addressData.Add**(**"ជ្រើសរើសខេត្ត"**)**

cbAddress.DataSource **=** addressData

cbAddress.SelectedIndex **=** cbAddress.Items.Count **-** **1**

End Sub

Private Sub cbCourse\_SelectedIndexChanged**(**sender As Object**,** e As EventArgs**)** Handles cbCourse.SelectedIndexChanged

If Integer.TryParse**(**cbCourse.SelectedValue.ToString**(),** Nothing**)** **=** False Then

Return

End If

If Integer.Parse**(**cbCourse.SelectedValue**)** **=** **-1** Then

cbTime.Enabled **=** False

Return

End If

Dim SubjectID As Integer **=** Convert.ToInt32**(**cbCourse.SelectedValue**)**

cbTime.DataSource **=** New BindingSource**(**register.GetTimeList**(**SubjectID**),** Nothing**)**

register.course.subject.GetSubjectByID**(**SubjectID**)**

txtPrice.Text **=** register.course.subject.basePrice.ToString**(**"F2"**)**

cbTime.DisplayMember **=** "Key"

cbTime.ValueMember **=** "Value"

cbTime.SelectedIndex **=** cbTime.Items.Count **-** **1**

cbTime.Enabled **=** True

End Sub

Private Sub btnRegister\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnRegister.Click

'Step 1 : Check Field

If Not CheckField**()** Then

Return

End If

'if don't Choose Student From List Thus Create a new student

If register.student.id **=** String.Empty Then

register.student.AddStudent**(**txtKhName.Text.Trim**(),** txtEngName.Text.Trim**(),** cbGender.Text**,** dtpDob.Text**,** cbAddress.Text.Trim**(),** txtPhone.Text.Trim**(),** SaveImageAndReturnPath**())**

End If

'Assign Course Data to Register

register.course.GetCourseByID**(**cbTime.SelectedValue.ToString**())**

'Check Student Duplicate Course or Schedule

If Not CheckValidation**()** Then

Return

End If

Dim Price **=** Convert.ToDecimal**(**txtPrice.Text**)**

Dim Dis **=** Convert.ToInt16**(**txtDis.Text**)**

register.discount **=** Price **\*** Dis **/** **100.0**

Dim amount As Decimal **=** Decimal.Parse**(**txtAmount.Text**)**

Dim unpaid As Decimal **=** amount **-** Convert.ToDecimal**(**txtPay.Text**)**

'Step 2 : Create Register

register.RegisterStudent**()**

register.payment.NewPayment**(**register.registerID**,** amount**,** unpaid**)** 'Create Payment

'Print Invoice

ExportAndShow**()**

'Step 4 Display Data

Display**()**

register **=** New Register**()**

btnClear.PerformClick**()**

End Sub

Private Sub cbTime\_SelectedIndexChanged**(**sender As Object**,** e As EventArgs**)** Handles cbTime.SelectedIndexChanged

If cbTime.Items.Count **=** **1** Then

Return

End If

If Not IsNumeric**(**cbTime.SelectedValue.ToString**())** Then

Return

End If

If cbTime.SelectedValue **=** **-1** Then

Return

End If

register.course.GetCourseByID**(**cbTime.SelectedValue.ToString**())**

txtRoom.Text **=** register.course.room.room

txtPrice.Text **=** register.course.subject.basePrice.ToString**(**"F2"**)**

txtAmount.Text **=** register.course.subject.basePrice.ToString**(**"F2"**)**

txtDis.Text **=** "0"

txtPay.Text **=** register.course.subject.basePrice.ToString**(**"F2"**)**

End Sub

Sub Regonize**()**

For Each col As DataGridViewColumn In DataGridView1.Columns

col.SortMode **=** DataGridViewColumnSortMode.NotSortable

Next

With DataGridView1

**.**ColumnHeadersHeight **=** **40**

**.**Columns**(0).**Width **=** **50** ' ID column width

**.**Columns**(0).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

**.**Columns**(0).**DefaultCellStyle.Format **=** "000"

**.**Columns**(1).**Width **=** **250** ' Course Name column width

**.**Columns**(2).**Width **=** **200** ' Description column width

**.**Columns**(3).**Width **=** **150** ' Description column width

**.**Columns**(3).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

**.**Columns**(4).**Width **=** **150**

**.**Columns**(5).**Width **=** **150**

**.**Columns**(6).**Width **=** **150**

**.**Columns**(0).**HeaderText **=** "លេខកូដ"

**.**Columns**(1).**HeaderText **=** "វគ្គសិក្សា"

**.**Columns**(2).**HeaderText **=** "គ្រូបង្រៀន"

**.**Columns**(3).**HeaderText **=** "ថ្ងៃចូលរៀន"

**.**Columns**(4).**HeaderText **=** "វេនសិក្សា"

**.**Columns**(5).**HeaderText **=** "បន្ទប់រៀន"

**.**Columns**(6).**HeaderText **=** "សិស្សសរុប"

**.**Columns**(4).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

**.**Columns**(5).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

**.**Columns**(6).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

**.**ClearSelection**()**

End With

End Sub

Private Sub PanelListShow\_Paint**(**sender As Object**,** e As PaintEventArgs**)** Handles PanelListShow.Paint

Display**()**

End Sub

Private Sub dtpDob\_TextChanged**(**sender As Object**,** e As EventArgs**)** Handles dtpDob.TextChanged

Dim input As String **=** dtpDob.Text.Replace**(**"/"**,** ""**)** ' Remove existing slashes

If input.Length **>=** **2** AndAlso input.Length **<** **4** Then

dtpDob.Text **=** input.Insert**(2,** "/"**)** ' Add slash after day

dtpDob.SelectionStart **=** dtpDob.Text.Length ' Move cursor to end

ElseIf input.Length **>=** **4** Then

dtpDob.Text **=** input.Insert**(2,** "/"**).**Insert**(5,** "/"**)** ' Add slash after month

dtpDob.SelectionStart **=** dtpDob.Text.Length ' Move cursor to end

End If

End Sub

Private Sub dtpDob\_Leave**(**sender As Object**,** e As EventArgs**)** Handles dtpDob.Leave

Dim inputDate As DateTime

If DateTime.TryParse**(**dtpDob.Text**,** inputDate**)** Then

dtpDob.Text **=** inputDate.ToString**(**"dd/MM/yyyy"**)**

Else

If dtpDob.Text.Length **=** **0** Then

Return

End If

MessageBox.Show**(**"សូមបញ្ចូលថ្ងៃខែឆ្នាំកំណើតរបស់សិស្ស"**,** "បញ្ចូលទិន្ន័យមិនត្រឹមត្រូវ"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

Return

End If

End Sub

Private Sub btnClear\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnClear.Click

txtKhName.Clear**()**

txtEngName.Clear**()**

txtDis.Text **=** "0"

txtPay.Clear**()**

txtPhone.Clear**()**

txtAmount.Text **=** "0"

txtPrice.Text **=** "0"

dtpDob.Clear**()**

txtRoom.Text **=** ""

cbAddress.SelectedIndex **=** cbAddress.Items.Count **-** **1**

cbCourse.SelectedIndex **=** cbCourse.Items.Count **-** **1**

cbGender.SelectedIndex **=** cbGender.Items.Count **-** **1**

cbTime.SelectedIndex **=** cbTime.Items.Count **-** **1**

cbTime.Enabled **=** False

picStudent.ImageLocation **=** Application.StartupPath **&** "\Images\default.png"

register **=** New Register**()**

End Sub

Private Sub txtDis\_TextChanged**(**sender As Object**,** e As EventArgs**)** Handles txtDis.TextChanged

If cbCourse.SelectedIndex **=** cbCourse.Items.Count **-** **1** Then

Return

End If

Dim dis As Decimal

If cbCourse.Text **=** String.Empty Then

txtDis.Text **=** **0**

Return

End If

If Not **(**Decimal.TryParse**(**txtDis.Text**,** dis**))** Or dis **<** **0** Or dis **>** **100** Then

MessageBox.Show**(**"សូមបញ្ចូលភាគរយបញ្ចុះតម្លៃអោយបានត្រឹមត្រូវ!!!"**,** "បញ្ចូលទិន្ន័យមិនត្រឹមត្រូវ"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Warning**)**

txtDis.Focus**()**

txtDis.SelectAll**()**

Return

End If

Dim amount As Decimal

Dim price As Decimal **=** Decimal.Parse**(**txtPrice.Text**)**

amount **=** price **-** price **\*** **(**dis **/** **100.0)** ' Ensures decimal division

txtAmount.Text **=** amount.ToString**(**"F2"**)** ' Formats output as 2 decimal places

amount.ToString**(**"F2"**)**

End Sub

Private Sub btnChooseStudent\_Click\_1**(**sender As Object**,** e As EventArgs**)** Handles btnChooseStudent.Click

Dim student As New StudentList**()**

If student.ShowDialog**()** **=** **2** Then

Return

End If

register.student **=** student.student

txtKhName.Text **=** register.student.khName

txtEngName.Text **=** register.student.engName

txtPhone.Text **=** register.student.phone

cbGender.SelectedIndex **=** cbGender.FindStringExact**(**register.student.gender**)**

dtpDob.Text **=** register.student.DateOfBirth.ToString**(**"dd/MM/yyyy"**)**

cbAddress.SelectedIndex **=** cbAddress.FindStringExact**(**register.student.address**)**

picStudent.ImageLocation **=** register.student.picture

End Sub

Private Sub ExportAndShow**()**

Dim rdReport As New ReportDocument

rdReport.Load**(**"Report\Invoice.rpt"**)**

Dim num As Integer **=** Convert.ToInt16**(**register.payment.paymentID**)**

rdReport.SetParameterValue**(**"PaymentID"**,** num.ToString**(**"0000"**))**

rdReport.SetParameterValue**(**"khName"**,** register.student.khName**)**

rdReport.SetParameterValue**(2,** register.student.engName**)**

rdReport.SetParameterValue**(3,** register.student.address**)**

rdReport.SetParameterValue**(4,** register.student.phone**)**

rdReport.SetParameterValue**(7,** register.registerID**)**

rdReport.SetParameterValue**(9,** register.course.subject.Subject**)**

rdReport.SetParameterValue**(8,** register.course.subject.basePrice**)**

rdReport.SetParameterValue**(5,** register.payment.amount**)**

rdReport.SetParameterValue**(6,** register.payment.unpaid**)**

rdReport.SetParameterValue**(10,** register.course.startDate**)**

rdReport.SetParameterValue**(11,** register.course.GetSchedule**())**

rdReport.SetParameterValue**(12,** register.course.room.room**)**

rdReport.PrintOptions.PaperSize **=** PaperSize.PaperA4

Dim exportPath As String **=** Application.StartupPath **&** "\Export\Invoices"

If Not Directory.Exists**(**exportPath**)** Then

Directory.CreateDirectory**(**exportPath**)**

End If

exportPath **=** **(**exportPath **+** "\"**)** **&** register.student.khName **&** "\_" **&** Date.Now**().**ToString**(**"dd-MMM-yyyy"**)** **&** ".pdf"

rdReport.ExportToDisk**(**ExportFormatType.PortableDocFormat**,** exportPath**)**

Process.Start**(**exportPath**)**

End Sub

Sub SearchData**()** Handles txtSearch.TextChanged

If startup Then

Return

End If

Dim query As String **=** "SELECT tblCourse.ID, tblSubject.Subject, tblTeacher.EngName, tblCourse.StartDate, tblSchedule.Schedule, tblRoom.Room, (Select COUNT(\*) FROM tblRegister WHERE tblCourse.ID = tblRegister.CourseID) As TotalStudent

From tblTeacher INNER Join **(**tblSchedule INNER Join **(**tblRoom INNER Join **(**tblSubject INNER Join **(**tblCourseStatus INNER Join tblCourse On tblCourseStatus.ID **=** tblCourse.StatusID**)** ON tblSubject.ID **=** tblCourse.SubjectID**)** ON tblRoom.ID **=** tblCourse.RoomID**)** ON tblSchedule.ID **=** tblCourse.ScheduleID**)** ON tblTeacher.ID **=** tblCourse.TeacherID

Where tblCourse.StatusID **<=** **2** AND tblSubject.Subject LIKE @subject

Order By tblCourse.ID**;**"

Dim subject As String **=** If**(**txtSearch.Text **=** ""**,** "%"**,** txtSearch.Text **&** "%"**)**

Dim cmd As New OleDbCommand**(**query**,** register.GetConnection**())**

cmd.Parameters.AddWithValue**(**"@subject"**,** subject**)**

DataGridView1.DataSource **=** register.ExecuteQuery**(**cmd**)**

Regonize**()**

End Sub

Private Sub RegisterForm\_Load**(**sender As Object**,** e As EventArgs**)** Handles MyBase.Load

startup **=** False

End Sub

Private Sub DataGridView1\_CellDoubleClick**(**sender As Object**,** e As DataGridViewCellEventArgs**)** Handles DataGridView1.CellDoubleClick

Dim courseID As String **=** DataGridView1.SelectedRows**(0).**Cells**(0).**Value

register.course.GetCourseByID**(**courseID**)**

cbCourse.SelectedIndex **=** cbCourse.FindStringExact**(**register.course.subject.Subject**)**

cbTime.SelectedIndex **=** cbTime.FindStringExact**(**register.course.GetSchedule**)**

End Sub

Private Sub txtDis\_Leave**(**sender As Object**,** e As EventArgs**)** Handles txtDis.Leave

If txtDis.Text.Length **=** **0** Then

txtDis.Text **=** "0"

End If

End Sub

Private Sub SearchData**(**sender As Object**,** e As EventArgs**)** Handles txtSearch.TextChanged

End Sub

Private Sub Button1\_Click**(**sender As Object**,** e As EventArgs**)** Handles Button1.Click

txtSearch.Clear**()**

End Sub

End Class

**5.TeacherForm**

Imports System.Data.OleDb

Imports System.IO

Public Class TeachersForm

Dim teacher As Teacher **=** New Teacher**()**

Dim startup As Boolean **=** True

Public Sub New**()**

' This call is required by the designer.

InitializeComponent**()**

' Add any initialization after the InitializeComponent() call.

End Sub

'Display all teachers in DataGridView

Sub Display**()**

DataGridView1.DataSource **=** teacher.GetTeacherData**()**

Regonize**()**

End Sub

'Check User Input

Private Function CheckField**()** As Boolean

Dim khName As String **=** txtKhName.Text

Dim engName As String **=** txtEngName.Text.Trim**()**

Dim Gender As String **=** cbGender.Text

Dim DoB As Date **=** Convert.ToDateTime**(**dtpDob.Text.Trim**())**

Dim Address As String **=** cbAddress.Text.Trim**()**

Dim Phone As String **=** txtPhone.Text.Trim**()**

Dim Email As String **=** txtEmail.Text.Trim**()**

' Validate inputs

If String.IsNullOrEmpty**(**khName**)** Then

MessageBox.Show**(**"សូមបញ្ចូល គោត្តនាម នាម"**,** "Validation Error"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

Return False

ElseIf String.IsNullOrEmpty**(**engName**)** Then

MessageBox.Show**(**"សូមបញ្ចូល ឈ្មោះឡាតាំង"**,** "Validation Error"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

Return False

ElseIf String.IsNullOrEmpty**(**Gender**)** Then

MessageBox.Show**(**"សូមជ្រើសភេទ"**,** "Validation Error"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

Return False

ElseIf String.IsNullOrEmpty**(**DoB.ToString**())** Then

MessageBox.Show**(**"សូមបញ្ចូលថ្ងៃខែឆ្មាំកំណើត."**,** "Validation Error"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

Return False

ElseIf String.IsNullOrEmpty**(**Address**)** Then

MessageBox.Show**(**"សូមបញ្ជូលអាស័យដ្ឋាន"**,** "Validation Error"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

Return False

ElseIf String.IsNullOrEmpty**(**Phone**)** Or IsNumeric**(**Phone**)** **=** False Then

MessageBox.Show**(**"សូមបញ្ចូលលេខទូរស័ព្ទ"**,** "Validation Error"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

Return False

ElseIf String.IsNullOrEmpty**(**Email**)** Or Not Email.Contains**(**"@"**)** Then

MessageBox.Show**(**"សូមបញ្ចូល Email"**,** "Validation Error"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

Return False

End If

Return True

End Function

Private Sub DataGridView1\_Click**(**sender As Object**,** e As EventArgs**)** Handles DataGridView1.CellClick

If btnEdit.Visible **=** False Then

Return

End If

Dim teacterID As String **=** DataGridView1.CurrentRow.Cells**(0).**Value.ToString**()**

teacher.GetTeacherByID**(**teacterID**)**

txtKhName.Text **=** teacher.KhName

txtEngName.Text **=** teacher.EngName

cbAddress.SelectedIndex **=** cbAddress.FindStringExact**(**teacher.Address**)**

dtpDob.Text **=** teacher.DoB.ToString**(**"dd/MM/yyyy"**)**

txtPhone.Text **=** teacher.Phone

cbGender.Text **=** teacher.Gender

txtEmail.Text **=** teacher.Email

If Not String.IsNullOrEmpty**(**teacher.Picture**)** Then

picTeacher.ImageLocation **=** teacher.Picture

Else

picTeacher.ImageLocation **=** Application.StartupPath **&** "\Images\default.png"

End If

End Sub

Private Sub btnChooseImage\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnChooseImage.Click

OpenPic.Filter **=** "Image Files|\*.jpg;\*.jpeg;\*.png;\*.bmp"

OpenPic.Title **=** "Select a Picture"

OpenPic.InitialDirectory **=** Environment.GetFolderPath**(**Environment.SpecialFolder.MyPictures**)**

If OpenPic.ShowDialog**()** **=** DialogResult.OK Then

Dim filePath As String **=** OpenPic.FileName

picTeacher.ImageLocation **=** filePath

End If

End Sub

Sub Regonize**()**

If DataGridView1.Columns.Count **=** **7** Then

With DataGridView1

**.**Columns**(0).**HeaderText **=** "លេខកូដ"

**.**Columns**(0).**Width **=** **100**

**.**Columns**(0).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

**.**Columns**(0).**DefaultCellStyle.Format **=** "000"

**.**Columns**(1).**Width **=** **250**

**.**Columns**(1).**HeaderText **=** "គោត្តនាម នាម"

**.**Columns**(2).**Width **=** **250**

**.**Columns**(2).**HeaderText **=** "ឈ្មោះឡាតាំង"

**.**Columns**(3).**Width **=** **200**

**.**Columns**(3).**HeaderText **=** "ថ្ងៃខែឆ្នាំកំណើត"

**.**Columns**(3).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

**.**Columns**(4).**Width **=** **200**

**.**Columns**(4).**HeaderText **=** "អាស័យដ្ឋាន"

**.**Columns**(5).**Width **=** **250**

**.**Columns**(5).**HeaderText **=** "Email"

**.**Columns**(6).**Width **=** **200**

**.**Columns**(6).**HeaderText **=** "លេខទូរស័ព្ទ"

**.**Columns**(6).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleLeft

End With

For i As Integer **=** **0** To DataGridView1.Rows.Count **-** **1**

If i Mod **2** **=** **1** Then

DataGridView1.Rows**(**i**).**DefaultCellStyle.BackColor **=** Color.FromArgb**(254,** **254,** **254)** ' Alternate row color

Else

DataGridView1.Rows**(**i**).**DefaultCellStyle.BackColor **=** Color.FromArgb**(245,** **250,** **253)**

End If

For Each col As DataGridViewColumn In DataGridView1.Columns

col.SortMode **=** DataGridViewColumnSortMode.NotSortable

Next

Next i

End If

DataGridView1.ClearSelection**()**

End Sub

Private Sub btnAdd\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnAdd.Click

Panel2.Visible **=** True

picTeacher.ImageLocation **=** Application.StartupPath **&** "\Images\default.png"

btnAddNew.Visible **=** True

btnEdit.Visible **=** False

End Sub

Private Sub btnEdit\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnEdit.Click

If Not CheckField**()** Then

Return

End If

Dim count **=** **0**

If txtKhName.Text **<>** teacher.KhName Then

count **+=** **1**

End If

If txtEngName.Text **<>** teacher.EngName Then

count **+=** **1**

End If

If Convert.ToDateTime**(**dtpDob.Text**)** **<>** teacher.DoB Then

count **+=** **1**

End If

If cbAddress.Text **<>** teacher.Address Then

count **+=** **1**

End If

If txtPhone.Text **<>** teacher.Phone Then

count **+=** **1**

End If

If cbGender.Text **<>** teacher.Gender Then

count **+=** **1**

End If

If txtEmail.Text **<>** teacher.Email Then

count **+=** **1**

End If

If count **=** **0** Then

MessageBox.Show**(**"មិនមានការផ្លាស់ប្តូរទេ។"**,** "No Changes"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Information**)**

Return

End If

teacher.KhName **=** txtKhName.Text

teacher.EngName **=** txtEngName.Text.Trim**()**

teacher.DoB **=** Convert.ToDateTime**(**dtpDob.Text**)**

teacher.Address **=** cbAddress.Text.Trim**()**

teacher.Phone **=** txtPhone.Text.Trim**()**

teacher.Gender **=** cbGender.Text

teacher.Email **=** txtEmail.Text.Trim**()**

teacher.Picture **=** picTeacher.ImageLocation

teacher.UpdateTeacher**()**

Display**()**

End Sub

Private Sub btnAddNew\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnAddNew.Click

If Not CheckField**()** Then

Return

Else

teacher.KhName **=** txtKhName.Text

teacher.EngName **=** txtEngName.Text.Trim**()**

teacher.DoB **=** Convert.ToDateTime**(**dtpDob.Text**)**

teacher.Address **=** cbAddress.Text

teacher.Phone **=** txtPhone.Text.Trim**()**

teacher.Gender **=** cbGender.Text

teacher.Email **=** txtEmail.Text.Trim**()**

teacher.Picture **=** SaveImageAndReturnPath**()**

teacher.AddTeacher**()**

End If

Display**()**

End Sub

Function SaveImageAndReturnPath**()** As String

If Path.GetFileName**(**picTeacher.ImageLocation**)** **=** "default.png" Then

Return picTeacher.ImageLocation

End If

Dim source As String **=** picTeacher.ImageLocation

Dim imagesDir As String **=** Path.Combine**(**Application.StartupPath**,** "Images"**)**

Dim destination As String **=** Path.Combine**(**imagesDir**,** Path.GetFileName**(**source**))**

Try

If Not Directory.Exists**(**imagesDir**)** Then

Directory.CreateDirectory**(**imagesDir**)**

End If

FileCopy**(**source**,** destination**)**

Return destination

Catch ex As Exception

MessageBox.Show**(**"Failed to copy image: " **&** ex.Message**)**

Return ""

End Try

End Function

Private Sub btnCancel\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnCancel.Click

Panel2.Visible **=** False

txtKhName.Text **=** ""

txtEngName.Text **=** ""

cbAddress.SelectedIndex **=** cbAddress.Items.Count **-** **1**

txtPhone.Text **=** ""

txtEmail.Text **=** ""

cbGender.SelectedText **=** cbGender.Items.Count **-** **1**

dtpDob.Text **=** DateTime.Now**().**ToString**(**"dd/MM/yyyy"**)**

picTeacher.ImageLocation **=** Application.StartupPath **&** "\Images\default.png"

btnAddNew.Visible **=** False

End Sub

Private Sub DataGridView1\_MouseClick**(**sender As Object**,** e As MouseEventArgs**)** Handles DataGridView1.MouseClick

Dim menu As New ContextMenuStrip**()**

menu.Font **=** New Font**(**"Khmer OS System"**,** **11)**

menu.Items.Add**(**"លម្អិត"**,** Nothing**,** AddressOf ShowDetails**)**

menu.Items.Add**(**"កែប្រែ"**,** Nothing**,** AddressOf EditClass**)**

menu.Items.Add**(**"លុប"**,** Nothing**,** AddressOf DeleteCourse**)**

menu.Items**(0).**BackColor **=** Color.FromArgb**(245,** **250,** **253)**

If e.Button **=** MouseButtons.Right Then

Dim hit As DataGridView.HitTestInfo **=** DataGridView1.HitTest**(**e.X**,** e.Y**)**

If hit.RowIndex **>=** **0** Then

Dim teacherID As String **=** DataGridView1.Rows**(**hit.RowIndex**).**Cells**(0).**Value.ToString**()**

teacher.GetTeacherByID**(**teacherID**)**

menu.Show**(**DataGridView1**,** e.Location**)**

End If

End If

End Sub

Private Sub DeleteCourse**()**

If DialogResult.Yes **=** MessageBox.Show**(**"តើអ្នកពិតជាចង់លុប គ្រូបង្រៀនឈ្មោះ " **&** teacher.KhName **&** " នេះ មែនទេ?"**,** "Confirm Delete"**,** MessageBoxButtons.YesNo**,** MessageBoxIcon.Question**)** Then

teacher.DeleteTeacher**(**teacher.TeacherID**)**

Display**()**

End If

End Sub

Private Sub ShowDetails**()**

Dim detailForm As New TeacherDetail**(**teacher**)**

detailForm.ShowDialog**()**

detailForm.Dispose**()**

Display**()**

End Sub

Private Sub EditClass**()**

Panel2.Visible **=** True

txtKhName.Text **=** teacher.KhName

txtEngName.Text **=** teacher.EngName

cbAddress.SelectedIndex **=** cbAddress.FindStringExact**(**teacher.Address**)**

txtPhone.Text **=** teacher.Phone

dtpDob.Text **=** teacher.DoB.ToString**(**"dd/MM/yyyy"**)**

cbGender.SelectedIndex **=** cbGender.FindStringExact**(**teacher.Gender**)**

txtEmail.Text **=** teacher.Email

picTeacher.ImageLocation **=** teacher.Picture

btnEdit.Visible **=** True

btnAddNew.Visible **=** False

End Sub

Sub SearchData**()** Handles cbSearchAddress.SelectedIndexChanged**,** txtSearch.TextChanged

If startup Then

Return

End If

Dim query As String **=** "SELECT tblTeacher.ID, tblTeacher.KhName AS KhmerName, tblTeacher.EngName AS EnglishName, tblTeacher.DOB AS DateOfBirth,

tblTeacher.Address AS PlaceOfBirth**,** tblTeacher.Email**,** tblTeacher.Phone FROM tblTeacher

WHERE Address LIKE @address AND **(**ID Like @id OR **(**KhName Like @name OR engName Like @name**));**"

Dim cmd As OleDbCommand **=** New OleDbCommand**(**query**,** teacher.GetConnection**())**

Dim name As String **=** If**(**txtSearch.Text.Trim**()** **=** ""**,** "%"**,** txtSearch.Text.Trim**()** **&** "%"**)**

Dim id As String **=** If**(**IsNumeric**(**txtSearch.Text.Trim**()),** txtSearch.Text.Trim**()** **&** "%"**,** "0"**)**

Dim address As String **=** If**(**cbSearchAddress.SelectedIndex **=** cbSearchAddress.Items.Count **-** **1,** "%"**,** cbSearchAddress.SelectedItem.ToString**())**

address **=** "%" **&** address **&** "%"

cmd.Parameters.AddWithValue**(**"@address"**,** address**)**

cmd.Parameters.AddWithValue**(**"@id"**,** id**)**

cmd.Parameters.AddWithValue**(**"@name"**,** name**)**

Dim dt As DataTable **=** teacher.ExecuteQuery**(**cmd**)**

DataGridView1.DataSource **=** dt

Regonize**()**

End Sub

Sub GetCbAddress**()**

Dim SearchAddressData As ArrayList **=** teacher.GetAddressData**()**

SearchAddressData.Add**(**"គ្រប់ខេត្តទាំងអស់"**)**

cbSearchAddress.DataSource **=** SearchAddressData

cbSearchAddress.SelectedIndex **=** cbSearchAddress.Items.Count **-** **1**

Dim addressData **=** teacher.GetAddressData**()**

addressData.Add**(**"ជ្រើសរើសខេត្ត"**)**

cbAddress.DataSource **=** addressData

cbAddress.SelectedIndex **=** cbAddress.Items.Count **-** **1**

End Sub

Private Sub btnClear\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnClear.Click

cbAddress.SelectedIndex **=** cbAddress.Items.Count **-** **1**

txtSearch.Text **=** ""

End Sub

Private Sub btnAdd\_Click\_1**(**sender As Object**,** e As EventArgs**)** Handles btnAdd.Click

Panel2.Visible **=** True

End Sub

Private Sub Panel9\_Paint\_1**(**sender As Object**,** e As PaintEventArgs**)** Handles Panel9.Paint

startup **=** False

End Sub

Private Sub Guna2Panel1\_Paint**(**sender As Object**,** e As PaintEventArgs**)** Handles Guna2Panel1.Paint

Display**()**

GetCbAddress**()**

End Sub

End Class

**6.StudentForm**

Imports System.Data.OleDb

Imports System.Windows.Forms.VisualStyles.VisualStyleElement

Public Class StudentForm

Dim student As Student **=** New Student**()**

Dim startup As Boolean **=** True ' To prevent initial search on form load

Public Sub New**()**

'Constructor for StudentForm class

InitializeComponent**()**

picStudent.ImageLocation **=** Application.StartupPath **&** "\Images\default.png"

GetCbAddress**()**

End Sub

Private Sub Display**()**

DataGridView1.DataSource **=** student.GetStudentData**()**

Regonize**()**

End Sub

'Check User Input

Private Function CheckField**()** As Boolean

Dim khName As String **=** txtKhName.Text.Trim**()**

Dim engName As String **=** txtEngName.Text.Trim**()**

Dim address As String **=** cbAddress.Text.Trim**()**

Dim phone As String **=** txtPhone.Text.Trim**()**

If String.IsNullOrEmpty**(**khName**)** Then

MessageBox.Show**(**"សូ​មបញ្ជូលឈ្មោះសិស្ស"**,** "Input Error"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

Return False

End If

If String.IsNullOrEmpty**(**engName**)** Then

MessageBox.Show**(**"សូ​មបញ្ជូលឈ្មោះឡាតាំង"**,** "Input Error"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

Return False

End If

If String.IsNullOrEmpty**(**address**)** Then

MessageBox.Show**(**"សូមបញ្ជូលអាសស័យដ្ឋាន"**,** "Input Error"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

Return False

End If

If String.IsNullOrEmpty**(**phone**)** Then

MessageBox.Show**(**"សូមបញ្ជូលលេខទូរស័ព្ទ"**,** "Input Error"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

Return False

End If

Dim result As Boolean **=** True

If Not IsNumeric**(**phone**)** Or phone.Length **<=** **8** Or phone.Length **>=** **11** Then

MessageBox.Show**(**"លេខទូរស័ព្ទត្រូវតែជាលេខ"**,** "Input Error"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

result **=** False

End If

Dim count As Integer **=** **0**

If student.khName **<>** khName Then

count **+=** **1**

End If

If student.engName **<>** engName Then

count **+=** **1**

End If

If student.address **<>** address Then

count **+=** **1**

End If

If student.phone **<>** phone Then

count **+=** **1**

End If

If student.gender **<>** cbGender.Text Then

count **+=** **1**

End If

If student.DateOfBirth **<>** Convert.ToDateTime**(**dtpDob.Text**)** Then

count **+=** **1**

End If

If count **=** **0** Then

MessageBox.Show**(**"មិនមានការផ្លាស់ប្តូរទេ"**,** "កែប្រែព័ត៌មានសិស្ស"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Information**)**

Return False

End If

Return result

End Function

Sub Regonize**()**

If DataGridView1.Columns.Count **=** **7** Then

With DataGridView1

**.**Columns**(0).**Width **=** **75**

**.**Columns**(0).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

**.**Columns**(0).**DefaultCellStyle.Format **=** "000"

**.**Columns**(1).**Width **=** **250**

**.**Columns**(1).**HeaderText **=** "គោត្តនាម នាម"

**.**Columns**(2).**Width **=** **200**

**.**Columns**(2).**HeaderText **=** "ឈ្មោះឡាតាំង"

**.**Columns**(3).**Width **=** **75**

**.**Columns**(3).**HeaderText **=** "ភេទ"

**.**Columns**(3).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

**.**Columns**(4).**Width **=** **200**

**.**Columns**(4).**HeaderText **=** "ថ្ងៃខែឆ្នាំកំណើត"

**.**Columns**(4).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

**.**Columns**(5).**Width **=** **150**

**.**Columns**(5).**HeaderText **=** "អាស័យដ្ឋាន"

**.**Columns**(5).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

**.**Columns**(6).**Width **=** **200**

**.**Columns**(6).**HeaderText **=** "លេខទូរស័ព្ទ"

**.**Columns**(6).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

End With

For i As Integer **=** **0** To DataGridView1.Rows.Count **-** **1**

If i Mod **2** **=** **1** Then

DataGridView1.Rows**(**i**).**DefaultCellStyle.BackColor **=** Color.FromArgb**(254,** **254,** **254)** ' Alternate row color

Else

DataGridView1.Rows**(**i**).**DefaultCellStyle.BackColor **=** Color.FromArgb**(245,** **250,** **253)**

End If

Next i

For Each col As DataGridViewColumn In DataGridView1.Columns

col.SortMode **=** DataGridViewColumnSortMode.NotSortable

Next

End If

DataGridView1.ClearSelection**()**

End Sub

Private Sub DataGridView1\_MouseClick**(**sender As Object**,** e As MouseEventArgs**)** Handles DataGridView1.MouseClick

Dim menu As New ContextMenuStrip**()**

menu.Font **=** New Font**(**"Khmer OS System"**,** **11)**

menu.Items.Add**(**"លម្អិត"**,** Nothing**,** AddressOf ShowDetails**)**

menu.Items.Add**(**"កែប្រែ"**,** Nothing**,** AddressOf EditStudent**)**

menu.Items.Add**(**"លុប"**,** Nothing**,** AddressOf DeleteStudent**)**

menu.Items**(0).**BackColor **=** Color.FromArgb**(245,** **250,** **253)**

If e.Button **=** MouseButtons.Right Then

Dim hit As DataGridView.HitTestInfo **=** DataGridView1.HitTest**(**e.X**,** e.Y**)**

If hit.RowIndex **>=** **0** Then

Dim studentID As String **=** DataGridView1.Rows**(**hit.RowIndex**).**Cells**(0).**Value.ToString**()**

student.GetStudentByID**(**Convert.ToInt32**(**studentID**))**

menu.Show**(**DataGridView1**,** e.Location**)**

End If

End If

End Sub

Private Sub DeleteStudent**()**

If DialogResult.Yes **=** MessageBox.Show**(**"តើអ្នកពិតជាចង់លុប សិស្សឈ្មោះ" **&** student.khName **&** " នេះ មែនទេ?"**,** "Confirm Delete"**,** MessageBoxButtons.YesNo**,** MessageBoxIcon.Question**)** Then

student.DeleteStudent**()**

Display**()**

End If

End Sub

Private Sub ShowDetails**()**

Dim detailForm As New StudentDetail**(**student**)**

detailForm.ShowDialog**()**

detailForm.Dispose**()**

Display**()**

End Sub

Private Sub EditStudent**()**

Panel2.Visible **=** True

txtKhName.Text **=** student.khName

txtEngName.Text **=** student.engName

cbAddress.SelectedIndex **=** cbAddress.FindStringExact**(**student.address**)**

txtPhone.Text **=** student.phone

dtpDob.Text **=** student.DateOfBirth.ToString**(**"dd/MM/yyyy"**)**

cbGender.SelectedIndex **=** cbGender.FindStringExact**(**student.gender**)**

picStudent.ImageLocation **=** student.picture

End Sub

Private Sub Button2\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnClear.Click

Panel2.Visible **=** False

txtKhName.Text **=** ""

txtEngName.Text **=** ""

cbAddress.Text **=** ""

txtPhone.Text **=** ""

dtpDob.Text **=** DateTime.Now**().**ToString**(**"dd/MM/yyyy"**)**

End Sub

Private Sub btnEdit\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnEdit.Click

If CheckField**()** Then

student.UpdateStudent**(**student.id**,** txtKhName.Text.Trim**(),** txtEngName.Text.Trim**(),** cbGender.SelectedItem.ToString**(),** dtpDob.Text**,** cbAddress.Text.Trim**(),** txtPhone.Text.Trim**(),** picStudent.ImageLocation**)**

MessageBox.Show**(**"ព័ត៌មានសិស្សបានកែប្រែដោយជោគជ័យ"**,** "Success"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Information**)**

Display**()**

End If

End Sub

Private Sub dtpDob\_TextChanged**(**sender As Object**,** e As EventArgs**)** Handles dtpDob.TextChanged

Dim input As String **=** dtpDob.Text.Replace**(**"/"**,** ""**)** ' Remove existing slashes

If input.Length **>=** **2** AndAlso input.Length **<** **4** Then

dtpDob.Text **=** input.Insert**(2,** "/"**)** ' Add slash after day

dtpDob.SelectionStart **=** dtpDob.Text.Length ' Move cursor to end

ElseIf input.Length **>=** **4** Then

dtpDob.Text **=** input.Insert**(2,** "/"**).**Insert**(5,** "/"**)** ' Add slash after month

dtpDob.SelectionStart **=** dtpDob.Text.Length ' Move cursor to end

End If

End Sub

Private Sub dtpDob\_Leave**(**sender As Object**,** e As EventArgs**)** Handles dtpDob.Leave

Dim inputDate As DateTime

If DateTime.TryParse**(**dtpDob.Text**,** inputDate**)** Then

dtpDob.Text **=** inputDate.ToString**(**"dd/MM/yyyy"**)**

Else

If dtpDob.Text.Length **=** **0** Then

Return

End If

MessageBox.Show**(**"សូមបញ្ចូលថ្ងៃខែឆ្នាំកំណើតរបស់សិស្ស"**,** "បញ្ចូលទិន្ន័យមិនត្រឹមត្រូវ"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

Return

End If

End Sub

Sub GetCbAddress**()**

Dim SearchAddressData As ArrayList **=** student.GetAddressData**()**

SearchAddressData.Add**(**"គ្រប់ខេត្តទាំងអស់"**)**

cbSearchAddress.DataSource **=** SearchAddressData

cbSearchAddress.SelectedIndex **=** cbSearchAddress.Items.Count **-** **1**

Dim addressData **=** student.GetAddressData**()**

addressData.Add**(**"ជ្រើសរើសខេត្ត"**)**

cbAddress.DataSource **=** addressData

cbAddress.SelectedIndex **=** cbAddress.Items.Count **-** **1**

End Sub

Private Sub Panel1\_Paint**(**sender As Object**,** e As PaintEventArgs**)** Handles Panel1.Paint

'GetCbAddress()

'startup = False

End Sub

Sub SearchData**()** Handles cbSearchAddress.SelectedIndexChanged**,** txtSearch.TextChanged

If startup Then

Return

End If

Dim query As String **=** "SELECT tblStudent.ID, tblStudent.KhName, tblStudent.EngName, tblStudent.Gender,

tblStudent.DateOfBirth**,** tblStudent.Address**,** tblStudent.Phone FROM tblStudent

WHERE Address LIKE @address AND **(**ID Like @id AND **(**KhName Like @name OR EngName Like @name**));**"

Dim cmd As OleDbCommand **=** New OleDbCommand**(**query**,** student.GetConnection**())**

Dim id As String **=** If**(**IsNumeric**(**txtSearch.Text.Trim**()),** txtSearch.Text.Trim**()** **&** "%"**,** "%"**)**

Dim name As String **=** If**(**txtSearch.Text.Trim**()** **=** ""**,** "%"**,** If**(**IsNumeric**(**txtSearch.Text.Trim**()),** "%"**,** "%" **&** txtSearch.Text.Trim**()** **&** "%"**))**

Dim address As String **=** If**(**cbSearchAddress.SelectedIndex **=** cbSearchAddress.Items.Count **-** **1,** "%"**,** cbSearchAddress.SelectedItem.ToString**())**

address **=** "%" **&** address **&** "%"

cmd.Parameters.AddWithValue**(**"@address"**,** address**)**

cmd.Parameters.AddWithValue**(**"@id"**,** id**)**

cmd.Parameters.AddWithValue**(**"@name"**,** name**)**

Dim dt As DataTable **=** student.ExecuteQuery**(**cmd**)**

DataGridView1.DataSource **=** dt

Regonize**()**

End Sub

Private Sub Button1\_Click**(**sender As Object**,** e As EventArgs**)** Handles Button1.Click

txtSearch.Text **=** ""

cbSearchAddress.SelectedIndex **=** cbSearchAddress.Items.Count **-** **1**

End Sub

Private Sub btnChooseImage\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnChooseImage.Click

OpenPic.Filter **=** "Image Files|\*.jpg;\*.jpeg;\*.png;\*.bmp"

OpenPic.Title **=** "Select a Picture"

OpenPic.InitialDirectory **=** Environment.GetFolderPath**(**Environment.SpecialFolder.MyPictures**)**

If OpenPic.ShowDialog**()** **=** DialogResult.OK Then

Dim filePath As String **=** OpenPic.FileName

picStudent.ImageLocation **=** filePath

End If

End Sub

Private Sub Guna2Panel3\_Paint**(**sender As Object**,** e As PaintEventArgs**)** Handles Guna2Panel3.Paint

Display**()**

End Sub

Private Sub StudentForm\_Load**(**sender As Object**,** e As EventArgs**)** Handles MyBase.Load

GetCbAddress**()**

startup **=** False

End Sub

End Class

**7.CourseForm**

Imports System.Data.OleDb

Imports System.Data.SqlTypes

Imports System.Windows

Public Class CourseForm

Dim course As Course **=** New Course**()**

Dim startup As Boolean **=** True

Public Sub New**()**

'This call is required by the designer.

InitializeComponent**()**

'Add any initialization after the InitializeComponent() call.

GetcbStatus**()**

GetcbSearch**()**

End Sub

'Check Validation

Function CheckValidation**()** As Boolean

If Not CheckField**()** Then

Return False

End If

Dim result As Boolean **=** True

Dim query As String **=** "SELECT COUNT(\*) FROM tblCourse WHERE ScheduleID = @ScheduleID AND RoomID = @RoomID AND StatusID <= 2;" 'Check if the room and schedule are already in use

Dim cmd As New OleDbCommand**(**query**,** course.GetConnection**())**

cmd.Parameters.AddWithValue**(**"@ScheduleID"**,** cbSchedule.SelectedValue**)**

cmd.Parameters.AddWithValue**(**"@RoomID"**,** cbRoom.SelectedValue**)**

Dim count As Integer **=** Convert.ToInt32**(**course.ExecuteScalar**(**cmd**))**

If count **>** **0** Then

MessageBox.Show**(**"បន្ទប់ និង វេនសិក្សា ត្រូវបានប្រើប្រាស់រួចហើយ។ សូមជ្រើសរើសវេនសិក្សា ឬ បន្ទប់ផ្សេងទៀត។"**,** "ទិន្ន័យមិនត្រឹមត្រូវ"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

result **=** False

End If

query **=** "SELECT COUNT(\*) FROM tblCourse WHERE ID = @ID;"

cmd **=** New OleDbCommand**(**query**,** course.GetConnection**())**

cmd.Parameters.AddWithValue**(**"@ID"**,** txtCourseID.Text.Trim**())**

count **=** Convert.ToInt32**(**course.ExecuteScalar**(**cmd**))**

If count **>** **0** Then

MessageBox.Show**(**"លេខកូដវគ្គសិក្សា នេះមានរួចហើយ."**,** "ទិន្ន័យមិនត្រឹមត្រូវ"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

result **=** False

End If

query **=** "SELECT COUNT(\*) FROM tblCourse WHERE ID = @subjectID AND StatusID <= 2 AND ScheduleID = @ScheduleID ;" 'Check if the subject and teacher are already assigned

cmd **=** New OleDbCommand**(**query**,** course.GetConnection**())**

cmd.Parameters.AddWithValue**(**"@subjectID"**,** cbSubject.SelectedValue**)**

cmd.Parameters.AddWithValue**(**"@ScheduleID"**,** cbSchedule.SelectedValue**)**

count **=** Convert.ToInt32**(**course.ExecuteScalar**(**cmd**))**

If count **>** **0** Then

MessageBox.Show**(**"វគ្គសិក្សា ត្រូវបានប្រើប្រាស់រួចហើយនៅក្នុងវេនសិក្សានេះ ។ សូមជ្រើសរើស វគ្គសិក្សាថ្មី។"**,** "ទិន្ន័យមិនត្រឹមត្រូវ"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

result **=** False

End If

query **=** "SELECT COUNT(\*) FROM tblCourse WHERE TeacherID = @TeacherID AND StatusID <= 2 AND ScheduleID = @ScheduleID ;" 'Check if the subject and teacher are already assigned

cmd **=** New OleDbCommand**(**query**,** course.GetConnection**())**

cmd.Parameters.AddWithValue**(**"@TeacherID"**,** cbTeacher.SelectedValue**)**

cmd.Parameters.AddWithValue**(**"@ScheduleID"**,** cbSchedule.SelectedValue**)**

count **=** Convert.ToInt32**(**course.ExecuteScalar**(**cmd**))**

If count **>** **0** Then

MessageBox.Show**(**"គ្រូបង្រៀននេះ ត្រូវបានប្រើប្រាស់រួចហើយនៅក្នុងវេនសិក្សានេះ ។ សូមជ្រើសរើស គ្រូបង្រៀនថ្មី ។"**,** "ទិន្ន័យមិនត្រឹមត្រូវ"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

result **=** False

End If

If dtpStartDate.Value.Date **<=** DateTime.Now**().**Date Then

MessageBox.Show**(**"សូមបញ្ចូលថ្ងៃចូលរៀនអោយបានត្រឹមត្រូវ !!!!"**,** "ទិន្ន័យមិនត្រឹមត្រូវ"**,** MessageBoxButton.OK**,** MessageBoxIcon.Warning**)**

End If

If dtpEndDate.Value.Date **<=** dtpStartDate.Value.Date.AddDays**(10)** Then

MessageBox.Show**(**"សូមបញ្ចូលថ្ងៃចូលរៀនអោយបានត្រឹមត្រូវ !!!!"**,** "ទិន្ន័យមិនត្រឹមត្រូវ"**,** MessageBoxButton.OK**,** MessageBoxIcon.Warning**)**

End If

Return result

End Function

'Check User Input

Private Function CheckField**()** As Boolean

Dim ClassName As String **=** txtCourseID.Text.Trim**()**

Dim TimeID As Integer **=** Convert.ToInt32**(**cbSchedule.SelectedValue**)**

Dim RoomID As Integer **=** Convert.ToInt32**(**cbRoom.SelectedValue**)**

Dim TeacherID As Integer **=** cbTeacher.SelectedValue

Dim CourseID As Integer **=** cbSubject.SelectedValue

course.startDate **=** dtpStartDate.Value

If String.IsNullOrEmpty**(**ClassName**)** Then

MessageBox.Show**(**"សូមបញ្ចូលលេខកូដវគ្គសិក្សា"**,** "បញ្ចូលទិន្ន័យមិនត្រឹមត្រូវ"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

Return False

End If

If TimeID **<=** **0** Or RoomID **<=** **0** Or TeacherID **<=** **0** Or CourseID **<=** **0** Then

MessageBox.Show**(**"សូមជ្រើសរើវេនសិក្សា បន្ទប់ គ្រូបង្រៀន និង វគ្គសិក្សា"**,** "បញ្ចូលទិន្ន័យមិនត្រឹមត្រូវ"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

Return False

End If

Return True

End Function

'GetData From Class ManageClass

Private Sub Display**()**

InsertData**(**course.GetCourseData**())**

End Sub

Sub GetCbTime**()**

Dim timeList **=** course.GetTimeList**()**

timeList.Add**(**"ជ្រើសរើសវេនសិក្សា"**,** **-1)**

cbSchedule.DataSource **=** New BindingSource**(**timeList**,** Nothing**)**

cbSchedule.DisplayMember **=** "Key"

cbSchedule.ValueMember **=** "Value"

cbSchedule.SelectedIndex **=** cbSchedule.Items.Count **-** **1**

End Sub

Sub GetCbRoom**()**

Dim RoomList **=** course.GetRoomList**()**

RoomList.Add**(**"ជ្រើសរើសបន្ទប់រៀន"**,** **-1)**

cbRoom.DataSource **=** New BindingSource**(**RoomList**,** Nothing**)**

cbRoom.DisplayMember **=** "Key"

cbRoom.ValueMember **=** "Value"

cbRoom.SelectedIndex **=** cbRoom.Items.Count **-** **1**

End Sub

Sub GetCbTeacher**()**

Dim teacherList **=** course.GetTeacherList**()**

teacherList.Add**(**"ជ្រើសរើសគ្រូបង្រៀន"**,** "-1"**)**

cbTeacher.DataSource **=** New BindingSource**(**teacherList**,** Nothing**)**

cbTeacher.DisplayMember **=** "Key"

cbTeacher.ValueMember **=** "Value"

cbTeacher.SelectedIndex **=** cbTeacher.Items.Count **-** **1**

End Sub

Sub GetcbCourse**()**

Dim courseList **=** course.GetCourseList**()**

courseList.Add**(**"ជ្រើសរើសវគ្គសិក្សា"**,** **-1)**

cbSubject.DataSource **=** New BindingSource**(**courseList**,** Nothing**)**

cbSubject.DisplayMember **=** "Key"

cbSubject.ValueMember **=** "Value"

cbSubject.SelectedIndex **=** cbSubject.Items.Count **-** **1** ' Select the last item (which is "All")

End Sub

Sub GetcbSearch**()**

Dim teacherList **=** course.GetTeacherList**()**

teacherList.Add**(**"គ្រប់គ្រូទាំងអស់"**,** "%"**)**

cbSearchTeacher.DataSource **=** New BindingSource**(**teacherList**,** Nothing**)**

cbSearchTeacher.DisplayMember **=** "Key"

cbSearchTeacher.ValueMember **=** "Value"

cbSearchTeacher.SelectedIndex **=** cbSearchTeacher.Items.Count **-** **1**

End Sub

Sub GetcbStatus**()**

Dim classStatusList **=** course.GetClassStatusList**()**

classStatusList.Add**(**"គ្រប់ស្ថានភាព"**,** **0)**

cbSearchStatus.DataSource **=** New BindingSource**(**classStatusList**,** Nothing**)**

cbSearchStatus.DisplayMember **=** "Key"

cbSearchStatus.ValueMember **=** "Value"

cbSearchStatus.SelectedIndex **=** cbSearchStatus.Items.Count **-** **1** ' Select the last item (which is "All")

End Sub

Private Sub btnNewClass\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnNewClass.Click

If Not CheckValidation**()** Then

Return

End If

course.ID **=** txtCourseID.Text.Trim**()**

course.scheduleID **=** Convert.ToInt32**(**cbSchedule.SelectedValue.ToString**())**

course.room.GetRoomByID**(**cbRoom.SelectedValue**)**

course.teacher.TeacherID **=** cbTeacher.SelectedValue.ToString**()**

course.subject.ID **=** cbSubject.SelectedValue.ToString**()**

course.startDate **=** dtpStartDate.Value

course.endDate **=** dtpEndDate.Value

course.AddCourse**()**

Display**()**

End Sub

Sub SearchCourseData**()** Handles txtSearch.TextChanged**,** cbSearchTeacher.SelectedIndexChanged**,** cbSearchStatus.SelectedIndexChanged

If startup Then

Return

Else

Dim textSearch **=** If**(**txtSearch.Text.Trim**()** **=** ""**,** "%"**,** txtSearch.Text.Trim**())**

Dim id As String **=** If**(**IsNumeric**(**textSearch**),** textSearch **&** "%"**,** "%"**)**

Dim subject As String **=** If**(**IsNumeric**(**textSearch**),** "%"**,** textSearch **&** "%"**)**

Dim StatusID As String **=** If**(**cbSearchStatus.SelectedValue **=** **0,** "%"**,** cbSearchStatus.SelectedValue.ToString**())**

Dim teacherSearch **=** cbSearchTeacher.SelectedValue.ToString**()**

Dim query As String **=** " SELECT tblCourse.ID, tblSubject.Subject, tblTeacher.KhName, tblRoom.Room,

tblSchedule.Schedule**,** tblCourse.CurrentEnrollment**,** tblCourseStatus.Status

FROM tblSchedule INNER JOIN **(**tblRoom INNER JOIN **(**tblTeacher

INNER JOIN **(**tblSubject INNER JOIN **(**tblCourseStatus INNER JOIN tblCourse

ON **(**tblCourseStatus.ID **=** tblCourse.StatusID**)** AND **(**tblCourseStatus.ID **=** tblCourse.StatusID**))**

ON tblSubject.ID **=** tblCourse.SubjectID**)** ON tblTeacher.ID **=** tblCourse.TeacherID**)**

ON tblRoom.ID **=** tblCourse.RoomID**)** ON tblSchedule.ID **=** tblCourse.ScheduleID

WHERE tblCourse.ID LIKE @id AND tblSubject.Subject LIKE @subject AND tblTeacher.ID LIKE @teacher AND tblCourse.StatusID LIKE @StatusID**;**"

Dim cmd As New OleDbCommand**(**query**,** course.GetConnection**())**

cmd.Parameters.AddWithValue**(**"@id"**,** id**)**

cmd.Parameters.AddWithValue**(**"@subject"**,** subject**)**

cmd.Parameters.AddWithValue**(**"@Teacher"**,** teacherSearch**)**

cmd.Parameters.AddWithValue**(**"@StatusID"**,** StatusID**)**

InsertData**(**course.ExecuteQuery**(**cmd**))**

End If

End Sub

Private Sub Button1\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnClear.Click

cbSearchTeacher.SelectedIndex **=** cbSearchTeacher.Items.Count **-** **1**

cbSearchStatus.SelectedIndex **=** cbSearchStatus.Items.Count **-** **1**

txtSearch.Text **=** ""

Display**()**

End Sub

'Insert DATA To DataGridView

Private Sub InsertData**(**data As DataTable**)**

DataGridView1.DataSource **=** data

Regonize**()**

End Sub

' Regonize DataGridView2

Sub Regonize**()**

If DataGridView1.Columns.Count **=** **7** Then

With DataGridView1

**.**Columns**(0).**HeaderText **=** "លេខកូដ"

**.**Columns**(0).**Width **=** **100**

**.**Columns**(0).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

**.**Columns**(0).**DefaultCellStyle.Format **=** "000"

**.**Columns**(1).**Width **=** **250**

**.**Columns**(1).**HeaderText **=** "វគ្គសិក្សា"

**.**Columns**(2).**Width **=** **200**

**.**Columns**(2).**HeaderText **=** "គ្រូបង្រៀន"

**.**Columns**(3).**Width **=** **150**

**.**Columns**(3).**HeaderText **=** "បន្ទប់"

**.**Columns**(3).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

**.**Columns**(4).**Width **=** **200**

**.**Columns**(4).**HeaderText **=** "វេនសិក្សា"

**.**Columns**(4).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

**.**Columns**(5).**Width **=** **150**

**.**Columns**(5).**HeaderText **=** "សិស្សសរុប"

**.**Columns**(5).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

**.**Columns**(6).**Width **=** **200**

**.**Columns**(6).**HeaderText **=** "ស្ថានភាព"

**.**Columns**(6).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

End With

For i As Integer **=** **0** To DataGridView1.Rows.Count **-** **1**

If i Mod **2** **=** **1** Then

DataGridView1.Rows**(**i**).**DefaultCellStyle.BackColor **=** Color.FromArgb**(254,** **254,** **254)** ' Alternate row color

Else

DataGridView1.Rows**(**i**).**DefaultCellStyle.BackColor **=** Color.FromArgb**(245,** **250,** **253)**

End If

Next i

End If

For Each col As DataGridViewColumn In DataGridView1.Columns

col.SortMode **=** DataGridViewColumnSortMode.NotSortable

Next

DataGridView1.ClearSelection**()**

End Sub

Private Sub ManageClassForm\_Load**(**sender As Object**,** e As EventArgs**)** Handles MyBase.Load 'Load the form and initialize components

GetCbTime**()**

GetCbRoom**()**

GetCbTeacher**()**

GetcbCourse**()**

End Sub

Private Sub btnTime\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnTime.Click 'Show the Schedule form

Dim ScheduleForm As New Schedule**()**

ScheduleForm.ShowDialog**()**

ScheduleForm.Dispose**()**

startup **=** True

GetCbTime**()**

GetcbStatus**()**

GetcbSearch**()**

startup **=** False

End Sub

Private Sub btnRoom\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnRoom.Click 'Show the Room form

Dim RoomForm As New RoomForm**()**

RoomForm.ShowDialog**()**

RoomForm.Dispose**()**

GetCbRoom**()**

GetcbSearch**()**

GetcbStatus**()**

End Sub

Private Sub btnAdd\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnAdd.Click

Panel2.Visible **=** True 'Show the panel for adding a new class

cbStatus.Enabled **=** False

End Sub

Private Sub btnCancel\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnCancel.Click

txtCourseID.Text **=** ""

cbSubject.SelectedIndex **=** cbSubject.Items.Count **-** **1**

cbRoom.SelectedIndex **=** cbRoom.Items.Count **-** **1**

cbTeacher.SelectedIndex **=** cbTeacher.Items.Count **-** **1**

cbSchedule.SelectedIndex **=** cbSchedule.Items.Count **-** **1**

dtpStartDate.Value **=** DateTime.Now

Panel2.Visible **=** False

btnEdit.Visible **=** False

txtCourseID.ReadOnly **=** False

btnNewClass.Visible **=** True 'Hide the panel for adding a new class

End Sub

Private Sub DataGridView2\_Sorted**(**sender As Object**,** e As EventArgs**)** Handles DataGridView1.Sorted

Regonize**()** 'Prepare the DataGridView after sorting

End Sub

Private Sub DataGridView1\_MouseClick**(**sender As Object**,** e As MouseEventArgs**)** Handles DataGridView1.MouseClick

Dim menu As New ContextMenuStrip**()**

menu.Font **=** New Font**(**"Khmer OS System"**,** **11)**

menu.Items.Add**(**"លម្អិត"**,** Nothing**,** AddressOf ShowDetails**)**

menu.Items.Add**(**"កែប្រែ"**,** Nothing**,** AddressOf EditClass**)**

menu.Items.Add**(**"លុប"**,** Nothing**,** AddressOf DeleteCourse**)**

menu.Items**(0).**BackColor **=** Color.FromArgb**(245,** **250,** **253)**

If e.Button **=** MouseButtons.Right Then

Dim hit As DataGridView.HitTestInfo **=** DataGridView1.HitTest**(**e.X**,** e.Y**)**

If hit.RowIndex **>=** **0** Then

Dim ClassID As String **=** DataGridView1.Rows**(**hit.RowIndex**).**Cells**(0).**Value.ToString**()**

course.GetCourseByID**(**ClassID**)**

menu.Show**(**DataGridView1**,** e.Location**)**

End If

End If

End Sb

Private Sub DeleteCourse**()**

If DialogResult.Yes **=** MessageBox.Show**(**"តើអ្នកពិតជាចង់លុបថ្នាក់រៀននេះមែនទេ?"**,** "លុបវគ្គសិក្សា"**,** MessageBoxButtons.YesNo**,** MessageBoxIcon.Question**)** Then

course.DeleteCourse**()**

Display**()**

End If

End Sub

Private Sub ShowDetails**()**

Dim classDetail As New CourseDetail**(**course**)**

classDetail.ShowDialog**()**

classDetail.Dispose**()**

Display**()**

startup **=** True

GetcbSearch**()**

GetcbStatus**()**

startup **=** False

End Sub

Private Sub EditClass**()**

txtCourseID.Text **=** course.ID

cbSubject.SelectedIndex **=** cbSubject.FindStringExact**(**course.subject.Subject**)**

cbTeacher.SelectedValue **=** course.teacher.TeacherID

cbRoom.SelectedValue **=** course.room.id

cbSchedule.SelectedValue **=** course.scheduleID

dtpStartDate.Value **=** course.startDate

dtpEndDate.Value **=** course.endDate

cbStatus.SelectedIndex **=** course.statusID **-** **1**

Panel2.Visible **=** True

btnEdit.Visible **=** True

btnNewClass.Visible **=** False

cbStatus.Enabled **=** True

End Sub

Private Sub btnEdit\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnEdit.Click

If txtCourseID.Text **=** course.ID And cbSubject.SelectedValue **=** course.subject.ID And cbRoom.SelectedValue **=** course.room.id And cbTeacher.SelectedValue **=** course.teacher.TeacherID And dtpStartDate.Value **=** course.startDate And dtpEndDate.Value **=** course.endDate And

cbSchedule.SelectedValue **=** course.scheduleID And cbStatus.SelectedIndex **=** course.statusID **-** **1** Then

MessageBox.Show**(**"សូមកែប្រែវគ្គសិក្សា"**,** "ទិន្ន័យមិនត្រឹមត្រូវ"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Information**)**

Return

End If

If Not CheckField**()** Then

Return

End If

course.ID **=** txtCourseID.Text.Trim**()**

course.scheduleID **=** Convert.ToInt32**(**cbSchedule.SelectedValue.ToString**())**

course.room.GetRoomByID**(**cbRoom.SelectedValue**)**

course.teacher.TeacherID **=** cbTeacher.SelectedValue.ToString**()**

course.subject.ID **=** cbSubject.SelectedValue.ToString**()**

course.startDate **=** dtpStartDate.Value

course.endDate **=** dtpEndDate.Value

course.statusID **=** cbStatus.SelectedIndex **+** **1**

txtCourseID.ReadOnly **=** True

course.UpdateCourse**()**

MessageBox.Show**(**"វគ្គសិក្សា ត្រូវបានកែប្រែដោយជោគជ័យ!"**,** "បច្ចុប្បន្នភាពវគ្គសិក្សា"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Information**)**

Display**()**

End Sub

Private Sub Panel1\_Paint**(**sender As Object**,** e As PaintEventArgs**)** Handles Panel1.Paint

Display**()**

startup **=** False

End Sub

End Class

**8.SubjectForm**

Imports System.Windows

Imports System.Windows.Controls

Public Class SubjectForm

Dim subject As New Subject**()**

Dim startup As Boolean **=** True

Sub Display**()**

DataGridView1.DataSource **=** subject.GetSubjectData**()**

Regonize**()**

End Sub

Function CheckField**()** As Boolean

Dim subject As String **=** txtSubject.Text.Trim**()**

Dim description As String **=** txtDes.Text.Trim**()**

Dim creditHours As String **=** txtCreditHours.Text.Trim**()**

Dim basePrice As String **=** txtBasePrice.Text.Trim**()**

If String.IsNullOrEmpty**(**subject**)** Then

MessageBox.Show**(**"សូមបញ្ចូលមុខវិជ្ជាអោយបានត្រឹមត្រូវ!!!!"**,** "បញ្ចូលទិន្ន័យមិនត្រឹមត្រូវ"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

Return False

End If

If String.IsNullOrEmpty**(**description**)** Then

MessageBox.Show**(**"សូមបញ្ចូលការពិពណ៌នាអំពីមុខវិជ្ជា!!!!"**,** "បញ្ចូលទិន្ន័យមិនត្រឹមត្រូវ"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

Return False

End If

If String.IsNullOrEmpty**(**creditHours**)** OrElse Not IsNumeric**(**creditHours**)** Then

MessageBox.Show**(**"សូមបញ្ចូលរយៈពេលសិក្សា!!!!"**,** "បញ្ចូលទិន្ន័យមិនត្រឹមត្រូវ"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

Return False

End If

If String.IsNullOrEmpty**(**basePrice**)** OrElse Not IsNumeric**(**basePrice**)** Then

MessageBox.Show**(**"សូមបញ្ចូលតម្លៃនៃមុខវិជ្ជានេះ!!!!"**,** "បញ្ចូលទិន្ន័យមិនត្រឹមត្រូវ"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Error**)**

Return False

End If

Return True

End Function

Private Sub btmCreate\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnCreate.Click

If Not CheckField**()** Then

Return

Else

'Check Duplicate Subject

Dim dt As DataTable **=** Me.subject.GetSubjectData**()**

For Each row As DataRow In dt.Rows

If txtSubject.Text.Trim**()** **=** row**(1).**ToString**()** Then

MsgBox**(**"មុខវិជ្ជានេះគឺមានរូចហើយ"**)**

Return

End If

Next

subject.AddSubject**(**txtSubject.Text.Trim**(),** txtDes.Text.Trim**(),** Convert.ToInt32**(**txtCreditHours.Text.Trim**()),** Convert.ToDecimal**(**txtBasePrice.Text.Trim**()))**

MessageBox.Show**(**"វគ្គសិក្សា ៖ " **&** txtSubject.Text **&** "ត្រូវបានបង្កើតបានដោយជោគជ័យ!!!!"**,** "Success"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Information**)**

End If

Display**()**

End Sub

Private Sub Panel5\_Paint**(**sender As Object**,** e As PaintEventArgs**)** Handles Panel5.Paint

Display**()**

startup **=** False

End Sub

Private Sub Button1\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnAdd.Click

Panel1.Visible **=** True

btnCreate.Visible **=** True

End Sub

Private Sub DataGridView1\_MouseClick**(**sender As Object**,** e As MouseEventArgs**)** Handles DataGridView1.MouseClick

Dim menu As New ContextMenuStrip**()**

menu.Font **=** New Font**(**"Khmer OS System"**,** **11)**

menu.Items.Add**(**"លម្អិត"**,** Nothing**,** AddressOf ShowDetails**)**

menu.Items.Add**(**"កែប្រែ"**,** Nothing**,** AddressOf EditCourse**)**

menu.Items.Add**(**"លុប"**,** Nothing**,** AddressOf DeleteCourse**)**

menu.Items**(0).**BackColor **=** Color.FromArgb**(245,** **250,** **253)**

If e.Button **=** MouseButtons.Right Then

Dim hit As DataGridView.HitTestInfo **=** DataGridView1.HitTest**(**e.X**,** e.Y**)**

If hit.RowIndex **>=** **0** Then

DataGridView1.Rows**(**hit.RowIndex**).**Selected **=** True

Dim subjectID As String **=** DataGridView1.Rows**(**hit.RowIndex**).**Cells**(0).**Value.ToString**()**

subject.GetSubjectByID**(**subjectID**)**

menu.Show**(**DataGridView1**,** e.Location**)**

End If

End If

End Sub

Private Sub DeleteCourse**()**

If DialogResult.Yes **=** MessageBox.Show**(**"តើអ្នកពិតជាចង់លុបវគ្គសិក្សានេះទេ?"**,** "Confirm Delete"**,** MessageBoxButtons.YesNo**,** MessageBoxIcon.Question**)** Then

subject.DeleteSubject**()**

MessageBox.Show**(**"វគ្គសិក្សា ៖ " **&** subject.Subject **&** " ត្រូវបានលុបដោយជោគជ័យ!"**,** "Success"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Information**)**

Display**()**

End If

End Sub

Private Sub EditCourse**()**

Panel1.Visible **=** True

btnEdit.Visible **=** True

txtSubject.Text **=** subject.Subject

txtDes.Text **=** subject.Description.ToString**()**

txtCreditHours.Text **=** subject.CreditHours.ToString**()**

txtBasePrice.Text **=** subject.basePrice.ToString**(**"F2"**)**

End Sub

Private Sub ShowDetails**()**

Dim courseDialog As New SubjectDetail**(**subject**)**

courseDialog.ShowDialog**()**

Display**()**

End Sub

Private Sub btnCancel\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnCancel.Click

txtBasePrice.Text **=** ""

txtSubject.Text **=** ""

txtDes.Text **=** ""

txtCreditHours.Text **=** ""

subject.ID **=** ""

subject.Subject **=** ""

subject.Description **=** ""

subject.CreditHours **=** **0**

subject.basePrice **=** **0D**

btnEdit.Visible **=** False

btnCreate.Visible **=** False

Panel1.Visible **=** False

End Sub

Private Sub txtSearch\_\_TextChanged**(**sender As Object**,** e As EventArgs**)** Handles txtSearch.TextChanged

If startup Then

Return

End If

Dim dt As New DataTable**()**

Dim query As String **=** "SELECT tblSubject.ID, tblSubject.Subject, tblSubject.Description, tblSubject.CreditHours, tblSubject.BasePrice,

**(**SELECT COUNT**(\*)** FROM tblCourse WHERE tblCourse.ID **=** tblSubject.ID**)** AS AllCourses**,**

**(**SELECT COUNT**(\*)** FROM tblCourse WHERE StatusID **=** **3** AND tblCourse.ID **=** tblSubject.ID**)** AS CompleteCourse

FROM tblSubject

WHERE **((**tblSubject.ID **=** @id**)** or **(**tblSubject.Subject LIKE @Condition**));**"

Dim cmd As New OleDb.OleDbCommand**(**query**,** subject.GetConnection**())**

Dim id As Integer

Dim condition As String **=** txtSearch.Text.Trim**()**

If IsNumeric**(**condition**)** Then

id **=** Integer.Parse**(**condition**)**

Else

id **=** **0**

End If

condition **=** condition **+** "%"

cmd.Parameters.AddWithValue**(**"@id"**,** id**)**

condition **=** condition **&** "%"

cmd.Parameters.AddWithValue**(**"@Condition"**,** condition**)**

DataGridView1.DataSource **=** subject.ExecuteQuery**(**cmd**)**

Regonize**()**

End Sub

Private Sub btnClear\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnClear.Click

txtSearch.Texts **=** ""

Display**()**

End Sub

Sub Regonize**()**

With DataGridView1

**.**ColumnHeadersHeight **=** **40**

**.**Columns**(0).**Width **=** **50** ' ID column width

**.**Columns**(0).**DefaultCellStyle.Format **=** "000"

**.**Columns**(1).**Width **=** **200** ' Course Name column width

**.**Columns**(2).**Width **=** **250** ' Description column width

**.**Columns**(3).**Width **=** **150** ' Description column width

**.**Columns**(4).**Width **=** **150**

**.**Columns**(5).**Width **=** **150**

**.**Columns**(6).**Width **=** **150**

**.**Columns**(0).**HeaderText **=** "ID"

**.**Columns**(1).**HeaderText **=** "វគ្គសិក្សា"

**.**Columns**(2).**HeaderText **=** "ការពិពណ៌នា"

**.**Columns**(3).**HeaderText **=** "រយៈពេល (ម៉ោង)"

**.**Columns**(3).**DefaultCellStyle.Format **=** "00ម៉ោង"

**.**Columns**(4).**HeaderText **=** "តម្លៃ ($)"

**.**Columns**(4).**DefaultCellStyle.Format **=** "C2"

**.**Columns**(5).**HeaderText **=** "ចំនួនថ្នាក់រៀន"

**.**Columns**(6).**HeaderText **=** "ចំនួនថ្នាក់បញ្ចប់"

End With

For i As Integer **=** **0** To DataGridView1.Rows.Count **-** **1**

If i Mod **2** **=** **1** Then

DataGridView1.Rows**(**i**).**DefaultCellStyle.BackColor **=** Color.FromArgb**(254,** **254,** **254)** ' Alternate row color

Else

DataGridView1.Rows**(**i**).**DefaultCellStyle.BackColor **=** Color.FromArgb**(245,** **250,** **253)**

End If

For j As Integer **=** **3** To DataGridView1.Columns.Count **-** **1**

DataGridView1.Rows**(**i**).**Cells**(**j**).**Style.Alignment **=** DataGridViewContentAlignment.MiddleCenter

Next j

Next i

For Each col As DataGridViewColumn In DataGridView1.Columns

col.SortMode **=** DataGridViewColumnSortMode.NotSortable

Next

DataGridView1.ClearSelection**()**

End Sub

Private Sub btnEdit\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnEdit.Click

If CheckField**()** Then

If txtSubject.Text **=** subject.Subject And txtDes.Text **=** subject.Description And Convert.ToDecimal**(**txtBasePrice.Text.Trim**())** **=** subject.basePrice And Convert.ToInt16**(**txtCreditHours.Text.Trim**())** **=** subject.CreditHours Then

MessageBox.Show**(**"មិនមានទិន្ន័យណាត្រូវបានកែប្រែទេ!!!"**)**

Else

subject.Subject **=** txtSubject.Text.Trim**()**

subject.basePrice **=** Convert.ToDecimal**(**txtBasePrice.Text.Trim**())**

subject.CreditHours **=** Convert.ToInt32**(**txtCreditHours.Text.Trim**())**

subject.Description **=** txtDes.Text.Trim**()**

subject.UpdateSubject**()**

MsgBox**(**"មុខវិជ្ជាត្រូវបានកែប្រែដោយជោគជ័យ"**)**

Display**()**

End If

End If

End Sub

End Class

**9.ReportForm**

Imports System.Drawing.Printing

Imports System.IO

Imports CrystalDecisions.CrystalReports.Engine

Imports CrystalDecisions.Shared

Imports CrystalDecisions.Windows.Forms

Imports Org.BouncyCastle.Asn1

Public Class ReportForm

Dim report As New Report**()**

Dim reportDocument As New ReportDocument

Private Sub ReportForm\_Load**(**sender As Object**,** e As EventArgs**)** Handles MyBase.Load

End Sub

Private Sub cbReport\_SelectedIndexChanged**(**sender As Object**,** e As EventArgs**)** Handles cbReport.SelectedIndexChanged

If cbReport.SelectedIndex **=** cbReport.Items.Count **-** **1** Then

crsReport.ReportSource **=** Nothing

Return

ElseIf cbReport.SelectedIndex **=** **0** Then

StudentReport**()**

ElseIf cbReport.SelectedIndex **=** **1** Then

TeacherReport**()**

ElseIf cbReport.SelectedIndex **=** **2** Then

CourseReport**()**

ElseIf cbReport.SelectedIndex **=** **3** Then

SubjectReport**()**

ElseIf cbReport.SelectedIndex **=** **4** Then

PaymentReport**()**

End If

End Sub

Sub TeacherReport**()**

reportDocument.Load**(**Application.StartupPath **&** "\Report\TeacherReport.rpt"**)**

reportDocument.Database.Tables**(**"tblTeacherReport"**).**SetDataSource**(**report.GetTeacherReport**())**

crsReport.ReportSource **=** reportDocument

End Sub

Sub StudentReport**()**

reportDocument.Load**(**Application.StartupPath **&** "\Report\StudentReport.rpt"**)**

reportDocument.Database.Tables**(**"tblStudentReport"**).**SetDataSource**(**report.GetStudentReport**())**

crsReport.ReportSource **=** reportDocument

End Sub

Sub CourseReport**()**

reportDocument.Load**(**Application.StartupPath **&** "\Report\CourseReport.rpt"**)**

reportDocument.Database.Tables**(**"tblCourseReport"**).**SetDataSource**(**report.GetCourseReport**())**

crsReport.ReportSource **=** reportDocument

End Sub

Sub SubjectReport**()**

reportDocument.Load**(**Application.StartupPath **&** "\Report\SubjectReport.rpt"**)**

reportDocument.Database.Tables**(**"tblSubjectReport"**).**SetDataSource**(**report.GetSubjectReport**())**

crsReport.ReportSource **=** reportDocument

End Sub

Sub PaymentReport**()**

reportDocument.Load**(**Application.StartupPath **&** "\Report\PaymentReport.rpt"**)**

reportDocument.Database.Tables**(**"tblPaymentReport"**).**SetDataSource**(**report.GetPaymentReport**())**

crsReport.ReportSource **=** reportDocument

End Sub

Private Sub btnPrint\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnPrint.Click

If cbReport.SelectedIndex **=** cbReport.Items.Count **-** **1** Then

MessageBox.Show**(**"សូមជ្រើសរើរបាយការណ៍ជាមុនសិន!!!!!"**,** "Warning"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Warning**)**

Return

End If

Try

Dim dlg As New PrintDialog**()**

dlg.AllowSomePages **=** True

dlg.PrinterSettings **=** New PrinterSettings**()**

If dlg.ShowDialog**()** **=** DialogResult.OK Then

reportDocument.PrintOptions.PrinterName **=** dlg.PrinterSettings.PrinterName

reportDocument.PrintOptions.PaperSize **=** CrystalDecisions.Shared.PaperSize.PaperA4

reportDocument.PrintOptions.PaperOrientation **=** PaperOrientation.Landscape

reportDocument.PrintToPrinter**(**dlg.PrinterSettings.Copies**,** False**,** **0,** **0)**

End If

Catch ex As Exception

MessageBox.Show**(**"របាយការណ៍័មិនអាចPrintបានទេ"**)**

End Try

End Sub

Private Sub Guna2Button2\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnExport.Click

If cbReport.SelectedIndex **=** cbReport.Items.Count **-** **1** Then

MessageBox.Show**(**"សូមជ្រើសរើសរបាយការណ៍!!!!"**,** "Warrning"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Warning**)**

Return

End If

Dim exportOption As ExportOptions

Dim diskFileDestinationOptions As New DiskFileDestinationOptions**()**

Dim exportPath As String **=** Application.StartupPath **&** "\Export\Report"

If Not Directory.Exists**(**exportPath**)** Then

Directory.CreateDirectory**(**exportPath**)**

End If

Dim sfd As New SaveFileDialog**()**

sfd.InitialDirectory **=** exportPath

sfd.Filter **=** "PDF Files|\*.pdf|Excel Files|\*.xls"

sfd.FileName **=** cbReport.Text **&** "\_" **&** DateTime.Now**().**ToString**(**"dd-MM-yyyy"**)**

If sfd.ShowDialog**()** **=** DialogResult.OK Then

diskFileDestinationOptions.DiskFileName **=** sfd.FileName

Else

Return

End If

exportOption **=** reportDocument.ExportOptions

exportOption.ExportDestinationType **=** ExportDestinationType.DiskFile

exportOption.ExportDestinationOptions **=** diskFileDestinationOptions

Select Case Path.GetExtension**(**sfd.FileName**).**ToLower**()**

Case ".pdf"

exportOption.ExportFormatType **=** ExportFormatType.PortableDocFormat

exportOption.FormatOptions **=** New PdfRtfWordFormatOptions**()**

Case ".xls"

exportOption.ExportFormatType **=** ExportFormatType.Excel

exportOption.FormatOptions **=** New ExcelFormatOptions**()** With **{**

**.**ExcelUseConstantColumnWidth **=** True

**}**

Case ".html"

exportOption.ExportFormatType **=** ExportFormatType.HTML40

exportOption.FormatOptions **=** New HTMLFormatOptions**()** With **{**

**.**HTMLBaseFolderName **=** Path.GetDirectoryName**(**sfd.FileName**),**

**.**HTMLFileName **=** Path.GetFileName**(**sfd.FileName**)**

**}**

Case Else

MessageBox.Show**(**"Unsupported export format."**)**

Return

End Select

reportDocument.Export**()**

Process.Start**(**sfd.FileName**)**

End Sub

End Class

**10.RoomForm**

Imports System.Data.OleDb

Public Class RoomForm

Dim conn As New ConnectionDB**()**

Dim room As New Room

Sub Display**()**

Dim query As String **=** "SELECT tblRoom.\* FROM tblRoom;"

Dim dt As DataTable **=** conn.ExecuteQuery**(**query**)**

DataGridView2.DataSource **=** dt

DataGridView2.Columns**(0).**Width **=** **100** ' ID column width)

DataGridView2.Columns**(1).**HeaderText **=** "ឈ្មោះបន្ទប់"

DataGridView2.ClearSelection**()**

For i As Integer **=** **0** To DataGridView2.Rows.Count **-** **1**

If i Mod **2** **=** **1** Then

DataGridView2.Rows**(**i**).**DefaultCellStyle.BackColor **=** Color.FromArgb**(254,** **254,** **254)** ' Alternate row color

Else

DataGridView2.Rows**(**i**).**DefaultCellStyle.BackColor **=** Color.FromArgb**(245,** **250,** **253)**

End If

Next i

End Sub

Private Sub Panel2\_Paint**(**sender As Object**,** e As PaintEventArgs**)** Handles Panel2.Paint

Display**()**

End Sub

Private Sub btnAdd\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnAdd.Click

If txtRoom.Text.Trim**()** **=** "" Then

MsgBox**(**"សូមបញ្ចូលឈ្មោះបន្ទប់"**)**

Return

End If

Dim query As String **=** "SELECT COUNT(\*) FROM tblRoom WHERE Room = @room;"

Dim cmd As New OleDbCommand**(**query**,** room.GetConnection**())**

cmd.Parameters.AddWithValue**(**"@room"**,** txtRoom.Text.Trim**())**

Dim count As Integer **=** Convert.ToInt16**(**room.ExecuteScalar**(**cmd**))**

If count **>** **0** Then

MessageBox.Show**(**"បន្ទប់សិកស្សានេះមានរួចហើយ"**,** "បន្ទប់មានរួចហើយ"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Warning**)**

Return

End If

query **=** "INSERT INTO tblRoom (Room) VALUES (@RoomName);"

cmd **=** New OleDbCommand**(**query**,** conn.GetConnection**())**

cmd.Parameters.AddWithValue**(**"@RoomName"**,** txtRoom.Text.Trim**())**

conn.ExecuteNonQuery**(**cmd**)**

txtRoom.Text **=** ""

MessageBox.Show**(**"បន្ទប់សិក្សា " **&** txtRoom.Text **&** " នេះត្រូវបានបង្កើតដោយជោគជ័យ"**,** "ជោគជ័យ"**,** MessageBoxButtons.OK**,** MessageBoxIcon.Information**)**

Display**()**

End Sub

Private Sub btnDelete\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnDelete.Click

If DialogResult.Yes **=** MessageBox.Show**(**"តើអ្នកពិតជាចង់លុបបន្ទប់នេះមែនទេ?"**,** "បញ្ជាក់"**,** MessageBoxButtons.YesNo**,** MessageBoxIcon.Question**)** Then

Dim query As String **=** "DELETE FROM tblRoom WHERE ID = @ID;"

Dim cmd As OleDbCommand **=** New OleDbCommand**(**query**,** conn.GetConnection**())**

cmd.Parameters.AddWithValue**(**"@ID"**,** room.id**)**

conn.ExecuteNonQuery**(**cmd**)**

Display**()**

End If

End Sub

Private Sub DataGridView2\_CellClick**(**sender As Object**,** e As DataGridViewCellEventArgs**)** Handles DataGridView2.CellClick

room.GetRoomByID**(**Convert.ToInt16**(**DataGridView2.CurrentRow.Cells**(0).**Value**))**

txtRoom.Text **=** room.room

End Sub

Private Sub btnClear\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnClear.Click

txtRoom.Clear**()**

End Sub

End Class

**11.ScheduleForm**

Imports System.Data.OleDb

Imports Org.BouncyCastle.Bcpg

Public Class Schedule

Dim conn As New ConnectionDB**()**

Dim id As Integer **=** **0**

Private Sub Schedule\_Load**(**sender As Object**,** e As EventArgs**)** Handles MyBase.Load

cbDay.SelectedIndex **=** cbDay.Items.Count **-** **1**

Display**()**

End Sub

Sub Display**()**

Dim query As String **=** "SELECT \* FROM tblSchedule"

Dim dt As DataTable **=** conn.ExecuteQuery**(**query**)**

DataGridView2.DataSource **=** dt

DataGridView2.Columns**(0).**Width **=** **100** ' ID column width

DataGridView2.ClearSelection**()**

For i As Integer **=** **0** To DataGridView2.Rows.Count **-** **1**

If i Mod **2** **=** **1** Then

DataGridView2.Rows**(**i**).**DefaultCellStyle.BackColor **=** Color.FromArgb**(254,** **254,** **254)** ' Alternate row color

Else

DataGridView2.Rows**(**i**).**DefaultCellStyle.BackColor **=** Color.FromArgb**(245,** **250,** **253)**

End If

Next i

End Sub

Private Sub btnAdd\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnAdd.Click

If cbDay.SelectedIndex **=** cbDay.Items.Count **-** **1** Then

MsgBox**(**"សូមជ្រើសរើសវេនសិក្សាជាមុនសិន"**)**

Return

End If

Dim schedule As String **=** cbDay.SelectedItem.ToString**()** **+** " " **+** tpStart.Value.ToString**(**"hh:mm tt"**).**Split**(**" "c**)(0)** **+** "-" **+** tpEnd.Value.ToString**(**"hh:mm tt"**)**

If IsDuplicate**(**schedule**)** Then

MsgBox**(**"សូមអភ័យទោសវេនសិក្សានេះមានរួចហើយ"**)**

Return

End If

Dim query As String **=** "INSERT INTO tblSchedule (Schedule) VALUES (@Schedule);"

Dim cmd As OleDbCommand **=** New OleDbCommand**(**query**,** conn.GetConnection**())**

cmd.Parameters.AddWithValue**(**"@Schedule"**,** schedule**)**

conn.ExecuteNonQuery**(**cmd**)**

Display**()**

End Sub

Private Sub btnDelete\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnDelete.Click

If DataGridView2.SelectedRows.Count **=** **0** Then

MsgBox**(**"សូមជ្រើសរើសទិន្ន័យពីបញ្ជីវេនសិក្សាជាមុនសិន !!!"**)**

Return

End If

Dim id As Integer **=** Convert.ToInt32**(**DataGridView2.CurrentRow.Cells**(0).**Value**)**

Dim query As String **=** "DELETE FROM tblSchedule WHERE ID = @ID;"

Dim cmd As OleDbCommand **=** New OleDbCommand**(**query**,** conn.GetConnection**())**

cmd.Parameters.AddWithValue**(**"@ID"**,** id**)**

conn.ExecuteNonQuery**(**cmd**)**

Display**()**

id **=** **0** ' Reset id after deletion

End Sub

Function IsDuplicate**(**value As String**)**

Dim query As String **=** "SELECT COUNT(\*) FROM tblSchedule WHERE Schedule = @Schedule;"

Dim cmd As OleDbCommand **=** New OleDbCommand**(**query**,** conn.GetConnection**())**

cmd.Parameters.AddWithValue**(**"@Schedule"**,** value**)**

conn.OpenConnection**()**

Dim count As Integer **=** Convert.ToInt32**(**cmd.ExecuteScalar**())**

conn.CloseConnection**()**

Return count **>** **0**

End Function

Private Sub DataGridView2\_CellClick**(**sender As Object**,** e As DataGridViewCellEventArgs**)** Handles DataGridView2.CellClick

Dim schedule**()** As String **=** DataGridView2.CurrentRow.Cells**(1).**Value.ToString**().**Split**(**" "**)**

cbDay.SelectedIndex **=** If**(**schedule**(0)** **=** cbDay.Items**(0).**ToString**(),** **0,** **1)** ' Weekday or Weekend

Dim time**()** As String **=** schedule**(1).**Split**(**"-"**)**

Dim parsedTime As DateTime **=** DateTime.Parse**(**time**(0)** **&** " " **&** schedule**(2))**

tpStart.Value **=** DateTime.Today.AddHours**(**parsedTime.Hour**)**

parsedTime **=** DateTime.Parse**(**time**(1)** **&** " " **&** schedule**(2))**

tpEnd.Value **=** DateTime.Today.AddHours**(**parsedTime.Hour**)**

End Sub

Private Sub btnClear\_Click**(**sender As Object**,** e As EventArgs**)** Handles btnClear.Click

cbDay.SelectedIndex **=** cbDay.Items.Count **-** **1**

DataGridView2.ClearSelection**()**

id **=** **0**

End Sub

End Class

**12.CrouseDetail Form**

Imports System.Data.OleDb

Public Class CourseDetail

Dim course As New Course**()**

Public Sub New**(**classDetail As Course**)**

' This call is required by the designer.

InitializeComponent**()**

' Add any initialization after the InitializeComponent() call.

Me.course **=** classDetail

FillClassDetails**()**

End Sub

Sub FillClassDetails**()**

txtID.Text **=** course.ID

txtCourse.Text **=** course.subject.Subject

txtTeacher.Text **=** course.teacher.KhName

txtStartDate.Text **=** course.startDate.ToString**(**"dd/MM/yyyy"**)**

txtEndDate.Text **=** course.endDate.ToString**(**"dd/MM/yyyy"**)**

txtRoom.Text **=** course.room.room

txtSchedule.Text **=** course.GetSchedule**()**

txtStatus.Text **=** course.GetStatus**()**

End Sub

Sub GetData**()**

Dim query As String **=** "SELECT tblStudent.ID, tblStudent.KhName, tblStudent.Phone, tblStudent.Address, tblPaymentStatus.Status

FROM **(**tblPaymentStatus INNER JOIN tblPayment ON tblPaymentStatus.ID **=** tblPayment.StatusID**)** INNER JOIN **(**tblStudent INNER JOIN tblRegister ON tblStudent.ID **=** tblRegister.StudentID**)** ON tblPayment.RegisterID **=** tblRegister.ID

WHERE **(((**tblRegister.CourseID**)=**" & course.ID & "**));**"

Dim dt As DataTable **=** course.ExecuteQuery**(**query**)**

DataGridView1.DataSource **=** dt

Regonize**()**

End Sub

Sub Regonize**()**

If DataGridView1.Columns.Count **=** **5** Then

DataGridView1.Columns**(0).**Width **=** **100**

DataGridView1.Columns**(0).**HeaderText **=** "StudentID"

DataGridView1.Columns**(0).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

DataGridView1.Columns**(1).**Width **=** **200**

DataGridView1.Columns**(1).**HeaderText **=** "ឈ្មោះ"

DataGridView1.Columns**(2).**Width **=** **200**

DataGridView1.Columns**(2).**HeaderText **=** "លេខទូរស័ព្ទ"

DataGridView1.Columns**(2).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

DataGridView1.Columns**(3).**Width **=** **200**

DataGridView1.Columns**(3).**HeaderText **=** "អាស័យដ្ឋាន"

DataGridView1.Columns**(3).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

DataGridView1.Columns**(4).**Width **=** **150**

DataGridView1.Columns**(4).**HeaderText **=** "ស្ថានភាពបង់ប្រាក់"

DataGridView1.Columns**(4).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

For Each col As DataGridViewColumn In DataGridView1.Columns

col.SortMode **=** DataGridViewColumnSortMode.NotSortable

Next

End If

DataGridView1.ClearSelection**()**

End Sub

Private Sub CourseDetail\_Load**(**sender As Object**,** e As EventArgs**)** Handles MyBase.Load

GetData**()**

End Sub

End Class

**13.StudentDetail Form**

Imports System.Windows.Controls

Public Class StudentDetail

Dim student As New Student**()**

Public Sub New**(**student As Student**)**

' This call is required by the designer.

InitializeComponent**()**

' Add any initialization after the InitializeComponent() call.

Me.student **=** student

FillStudentDetails**()**

Display**()**

End Sub

Sub FillStudentDetails**()**

txtID.Text **=** student.id

txtkhName.Text **=** student.khName

txtEngName.Text **=** student.engName

txtAddress.Text **=** student.address

txtPhone.Text **=** student.phone

txtDob.Text **=** student.DateOfBirth.ToString**(**"dd/MM/yyyy"**)**

txtGender.Text **=** student.gender

picStudent.ImageLocation **=** student.picture

End Sub

Sub Display**()**

GetData**()**

With DataGridView1

**.**Columns**(0).**HeaderText **=** "លេខកូដវគ្គសិក្សា"

**.**Columns**(0).**Width **=** **60**

**.**Columns**(0).**DefaultCellStyle.Format **=** "000"

**.**Columns**(0).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

**.**Columns**(1).**Width **=** **220**

**.**Columns**(1).**HeaderText **=** "វគ្គសិក្សា"

**.**Columns**(2).**Width **=** **70**

**.**Columns**(2).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

**.**Columns**(2).**HeaderText **=** "ប្រាក់មិនទាន់បង"

**.**Columns**(2).**DefaultCellStyle.Format **=** "C2" ' Format as currency

**.**Columns**(3).**HeaderText **=** "ស្ថានភាពវគ្គសិក្សា"

**.**Columns**(3).**Width **=** **180**

**.**Columns**(3).**DefaultCellStyle.Alignment **=** DataGridViewContentAlignment.MiddleCenter

For Each col As DataGridViewColumn In **.**Columns

col.SortMode **=** DataGridViewColumnSortMode.NotSortable

Next

End With

End Sub

Sub GetData**()**

Dim query As String **=** "SELECT tblCourse.ID, tblSubject.Subject, tblPayment.Unpaid, tblCourseStatus.Status

FROM **(**tblCourseStatus INNER JOIN **(**tblSubject INNER JOIN tblCourse ON tblSubject.ID **=** tblCourse.SubjectID**)** ON **(**tblCourseStatus.ID **=** tblCourse.StatusID**)** AND **(**tblCourseStatus.ID **=** tblCourse.StatusID**))** INNER JOIN **(**tblPayment INNER JOIN tblRegister ON tblPayment.RegisterID **=** tblRegister.ID**)** ON tblCourse.ID **=** tblRegister.CourseID

WHERE **(((**tblRegister.StudentID**)=**" & student.id & "**));**"

Dim dt As DataTable **=** student.ExecuteQuery**(**query**)**

DataGridView1.DataSource **=** dt

End Sub

End Class

**14.StudentList Form**

Imports System.Data.OleDb

Imports System.Web.UI.WebControls

Public Class StudentList

Public student As New Student()

Dim startup As Boolean = True

Public Sub GetData() Handles txtSearch.TextChanged

Dim query As String = "SELECT tblStudent.ID, tblStudent.KhName,tblStudent.Gender,

tblStudent.DateOfBirth,tblStudent.Phone FROM tblStudent

WHERE (ID Like @id OR KhName Like @name);"

Dim cmd As New OleDbCommand(query, student.GetConnection())

Dim name As String = txtSearch.Text.Trim()

Dim id As String = If(name = "", "0", If(IsNumeric(name), name, "0")) & "%"

name = If(name = "", "%", "%" & name & "%")

cmd.Parameters.AddWithValue("@id", id)

cmd.Parameters.AddWithValue("@name", name)

DataGridView1.DataSource = student.ExecuteQuery(cmd)

If Not startup Then

Regonize()

End If

End Sub

Sub Regonize()

DataGridView1.Columns(0).HeaderText = "លេខកូដ"

DataGridView1.Columns(0).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter

DataGridView1.Columns(0).Width = 50

DataGridView1.Columns(0).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter

DataGridView1.Columns(1).HeaderText = "គោត្តនាម នាម"

DataGridView1.Columns(1).Width = 250

DataGridView1.Columns(1).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleLeft

DataGridView1.Columns(2).HeaderText = "ភេទ"

DataGridView1.Columns(2).Width = 80

DataGridView1.Columns(2).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter

DataGridView1.Columns(3).HeaderText = "ថ្ងៃខែឆ្នាំកំណើត"

DataGridView1.Columns(3).Width = 100

DataGridView1.Columns(3).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter

DataGridView1.Columns(4).HeaderText = "លេខទូរស័ព្ទ"

DataGridView1.Columns(4).Width = 150

DataGridView1.Columns(4).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleLeft

For i = 0 To DataGridView1.Columns.Count - 1

DataGridView1.Columns(i).SortMode = DataGridViewColumnSortMode.NotSortable

Next i

End Sub

Private Sub DataGridView1\_CellDoubleClick(sender As Object, e As DataGridViewCellEventArgs)

Dim id As Int16 = Convert.ToInt16(DataGridView1.SelectedRows(0).Cells(0).Value)

student.GetStudentByID(id)

Me.DialogResult = 1

Me.Close()

End Sub

Private Sub btnClear\_Click(sender As Object, e As EventArgs)

txtSearch.Clear()

End Sub

Private Sub StudentList\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

GetData()

startup = False

Regonize()

End Sub

End Class

**15.SubjectDetail From**

Public Class SubjectDetail

Dim subject As New Subject()

Public Sub New(course As Subject)

' This call is required by the designer.

InitializeComponent()

' Add any initialization after the InitializeComponent() call.

Me.subject = course

FillCourseDetails()

Display()

End Sub

Sub Display()

Dim query As String = "SELECT tblCourse.ID, tblTeacher.KhName AS Teacher, tblRoom.Room AS Room, tblSchedule.Schedule, tblCourse.CurrentEnrollment, tblCourseStatus.Status

FROM tblTeacher INNER JOIN (tblSchedule INNER JOIN (tblRoom INNER JOIN (tblCourseStatus INNER JOIN tblCourse ON tblCourseStatus.ID = tblCourse.StatusID) ON tblRoom.ID = tblCourse.RoomID) ON tblSchedule.ID = tblCourse.ScheduleID) ON tblTeacher.ID = tblCourse.TeacherID

WHERE tblCourse.SubjectID = @ID;"

Dim cmd As New OleDb.OleDbCommand(query, subject.GetConnection())

cmd.Parameters.AddWithValue("@ID", subject.ID)

DataGridView1.DataSource = subject.ExecuteQuery(cmd)

DataGridView1.ColumnHeadersHeight = 40

'DataGridView1.AllowUserToResizeColumns = True

'Customize DataGridView columns

DataGridView1.Columns(0).Width = 75

DataGridView1.Columns(1).Width = 200

DataGridView1.Columns(2).Width = 100

DataGridView1.Columns(3).Width = 200

DataGridView1.Columns(4).Width = 100

DataGridView1.Columns(5).Width = 100

DataGridView1.Columns(0).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter

DataGridView1.Columns(1).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleLeft

DataGridView1.Columns(2).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter

DataGridView1.Columns(3).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter

DataGridView1.Columns(4).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter

DataGridView1.Columns(5).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter

DataGridView1.Columns(0).HeaderText = "ClassID"

DataGridView1.Columns(1).HeaderText = "គ្រូបង្រៀន"

DataGridView1.Columns(2).HeaderText = "បន្ទប់"

DataGridView1.Columns(3).HeaderText = "វេនសិក្សា"

DataGridView1.Columns(4).HeaderText = "សិស្សសរុប"

DataGridView1.Columns(5).HeaderText = "ស្ថានភាព"

For i As Integer = 0 To DataGridView1.Rows.Count - 1

If i Mod 2 = 1 Then

DataGridView1.Rows(i).DefaultCellStyle.BackColor = Color.FromArgb(254, 254, 254) ' Alternate row color

Else

DataGridView1.Rows(i).DefaultCellStyle.BackColor = Color.FromArgb(245, 250, 253)

End If

For j As Integer = 3 To DataGridView1.Columns.Count - 1

DataGridView1.Rows(i).Cells(j).Style.Alignment = DataGridViewContentAlignment.MiddleCenter

Next j

Next i

For Each col As DataGridViewColumn In DataGridView1.Columns

col.SortMode = DataGridViewColumnSortMode.NotSortable

Next

DataGridView1.ClearSelection()

End Sub

Sub FillCourseDetails()

txtSubject.Text = subject.Subject

txtDes.Text = subject.Description.ToString()

txtDuration.Text = subject.CreditHours.ToString()

txtBasePrice.Text = subject.basePrice.ToString("F2")

End Sub

End Class

**16.TeacherDetail Form**

Public Class TeacherDetail

Dim teacher As Teacher = New Teacher()

Public Sub New(teacher As Teacher)

' This call is required by the designer.

InitializeComponent()

' Add any initialization after the InitializeComponent() call.

Me.teacher = teacher

End Sub

Private Sub btnClose\_Click(sender As Object, e As EventArgs)

Me.Close()

End Sub

Private Sub FillData()

txtID.Text = teacher.TeacherID

txtkhName.Text = teacher.KhName

txtEngName.Text = teacher.EngName

txtGender.Text = teacher.Gender

txtAddress.Text = teacher.Address

txtDob.Text = teacher.DoB

txtPhone.Text = teacher.Phone

txtEmail.Text = teacher.Email

picTeacher.ImageLocation = teacher.Picture

End Sub

Sub Display()

Dim query As String = "SELECT tblCourse.ID, tblSubject.Subject, tblSchedule.Schedule, tblRoom.Room, tblCourse.CurrentEnrollment, tblCourseStatus.Status

FROM tblSchedule INNER JOIN (tblRoom INNER JOIN ((tblSubject INNER JOIN (tblCourseStatus INNER JOIN tblCourse ON tblCourseStatus.ID = tblCourse.StatusID) ON tblSubject.ID = tblCourse.SubjectID) INNER JOIN tblRegister ON tblCourse.ID = tblRegister.CourseID) ON tblRoom.ID = tblCourse.RoomID) ON tblSchedule.ID = tblCourse.ScheduleID

GROUP BY tblCourse.ID, tblSubject.Subject, tblSchedule.Schedule, tblRoom.Room, tblCourse.CurrentEnrollment, tblCourseStatus.Status, tblCourse.TeacherID

HAVING (((tblCourse.TeacherID)=[@teacherID]));"

Dim cmd As New OleDb.OleDbCommand(query, teacher.GetConnection())

cmd.Parameters.AddWithValue("@teacherID", teacher.TeacherID)

DataGridView1.DataSource = teacher.ExecuteQuery(cmd)

Regonize()

End Sub

Private Sub Panel3\_Paint(sender As Object, e As PaintEventArgs) Handles Panel3.Paint

FillData()

Display()

End Sub

Sub Regonize()

' Customize DataGridView columns

DataGridView1.ColumnHeadersHeight = 40

DataGridView1.Columns(0).Width = 75

DataGridView1.Columns(0).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter

DataGridView1.Columns(1).Width = 250

DataGridView1.Columns(1).HeaderText = "វគ្គសិក្សា"

DataGridView1.Columns(2).Width = 200

DataGridView1.Columns(2).HeaderText = "វេនសិក្សា"

DataGridView1.Columns(3).Width = 100

DataGridView1.Columns(3).HeaderText = "បន្ទប់"

DataGridView1.Columns(3).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter

DataGridView1.Columns(4).Width = 100

DataGridView1.Columns(4).HeaderText = "សិស្សសរុប"

DataGridView1.Columns(4).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter

DataGridView1.Columns(5).Width = 100

DataGridView1.Columns(5).HeaderText = "ស្ថានភាព"

DataGridView1.Columns(5).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter

For i As Integer = 0 To DataGridView1.Rows.Count - 1

If i Mod 2 = 1 Then

DataGridView1.Rows(i).DefaultCellStyle.BackColor = Color.FromArgb(254, 254, 254) ' Alternate row color

Else

DataGridView1.Rows(i).DefaultCellStyle.BackColor = Color.FromArgb(245, 250, 253)

End If

Next i

For Each col As DataGridViewColumn In DataGridView1.Columns

col.SortMode = DataGridViewColumnSortMode.NotSortable

Next

DataGridView1.ClearSelection()

End Sub

End Class

**17.ConnectionDB Class**

Imports System.Data.OleDb

Public Class ConnectionDB

Private connstring As String = "Provider=Microsoft.ACE.OLEDB.16.0;Data Source=Data\StudentManagementSystem.accdb;Persist Security Info=False;"

Private conn As OleDbConnection

Public Sub New()

conn = New OleDbConnection(connstring)

End Sub

Public Function GetConnection() As OleDbConnection

If conn Is Nothing Then

conn = New OleDbConnection(connstring)

End If

Return conn

End Function

Public Sub OpenConnection()

If conn IsNot Nothing AndAlso conn.State = ConnectionState.Closed Then

conn.Open()

End If

End Sub

Public Sub CloseConnection()

If conn IsNot Nothing AndAlso conn.State = ConnectionState.Open Then

conn.Close()

End If

End Sub

Public Function ExecuteQuery(query As String) As DataTable

Dim dt As New DataTable()

Try

OpenConnection()

Dim cmd As New OleDbCommand(query, conn)

Dim adapter As New OleDbDataAdapter(cmd)

adapter.Fill(dt)

cmd.Dispose()

Catch ex As Exception

MsgBox("Error executing query: " & ex.Message)

Finally

CloseConnection()

End Try

Return dt

End Function

Public Function ExecuteQuery(cmd As OleDbCommand) As DataTable

Dim dt As New DataTable()

Try

OpenConnection()

Dim adapter As New OleDbDataAdapter(cmd)

adapter.Fill(dt)

cmd.Dispose()

Catch ex As Exception

MsgBox("Error executing query: " & ex.Message)

Finally

CloseConnection()

End Try

Return dt

End Function

Public Sub ExecuteNonQuery(query As String)

Try

OpenConnection()

Dim cmd As New OleDbCommand(query, conn)

cmd.ExecuteNonQuery()

cmd.Dispose()

Catch ex As Exception

Throw New Exception("Error executing non-query: " & ex.Message)

Finally

CloseConnection()

End Try

End Sub

Public Function ExecuteScalar(query As String) As Object

Dim result As Object = Nothing

Try

OpenConnection()

Dim cmd As New OleDbCommand(query, conn)

result = cmd.ExecuteScalar()

cmd.Dispose()

Catch ex As Exception

Throw New Exception("Error executing scalar: " & ex.Message)

Finally

CloseConnection()

End Try

Return result

End Function

Public Function ExecuteScalar(cmd As OleDbCommand) As Object

Dim result As Object = Nothing

Try

OpenConnection()

result = cmd.ExecuteScalar()

cmd.Dispose()

Catch ex As Exception

Throw New Exception("Error executing scalar: " & ex.Message)

Finally

CloseConnection()

End Try

Return result

End Function

Public Sub ExecuteNonQuery(cmd As OleDbCommand)

Try

OpenConnection()

cmd.ExecuteNonQuery()

cmd.Dispose()

Catch ex As Exception

MsgBox("Error executing non-query: " & ex.Message)

Finally

CloseConnection()

End Try

End Sub

End Class

**18. User Class**

Imports System.Data.OleDb

Public Class User

Public Shared userID As String

Public Shared userName As String

Public Shared password As String

Public Shared role As String

Public Shared conn As OleDbConnection

Public Shared connString As String = "Provider=Microsoft.ACE.OLEDB.16.0;Data Source=Data\StudentManagementSystem.accdb;Persist Security Info=False;"

Public Shared Function CheckLogin(username As String, password As String) As Boolean

conn = New OleDbConnection(connString)

conn.Open()

Dim query As String = "SELECT \* FROM tblUser WHERE UserName = @UserName;"

Dim cmd As OleDbCommand = New OleDbCommand(query, conn)

cmd.Parameters.AddWithValue("@UserName", username)

cmd.Parameters.AddWithValue("@Password", password)

Dim reader As OleDbDataReader = cmd.ExecuteReader()

If reader.Read() Then

userID = reader("ID").ToString()

username = reader("UserName").ToString()

role = reader("Role").ToString()

If reader("Password") = password Then

Return True

Else

Return False

End If

reader.Close()

conn.Close()

Else

reader.Close()

conn.Close()

Return False

End If

End Function

End Class

**19.Course Class**

Imports System.Data.OleDb

Imports System.Windows.Media.Animation

Public Class Course

Inherits ConnectionDB

Public ID As String

Public subject As Subject

Public teacher As Teacher

Public startDate As Date

Public endDate As Date

Public room As Room

Public scheduleID As Integer

Public statusID As Integer

Public Sub New()

' Constructor for ManageClass class

MyBase.New()

subject = New Subject()

teacher = New Teacher()

room = New Room()

End Sub

Public Function GetCourseData() As DataTable

Dim query As String = "SELECT tblCourse.ID, tblSubject.Subject, tblTeacher.KhName, tblRoom.Room,

tblSchedule.Schedule, tblCourse.CurrentEnrollment, tblCourseStatus.Status

FROM tblSchedule INNER JOIN (tblRoom INNER JOIN (tblTeacher

INNER JOIN (tblSubject INNER JOIN (tblCourseStatus INNER JOIN tblCourse

ON (tblCourseStatus.ID = tblCourse.StatusID) AND (tblCourseStatus.ID = tblCourse.StatusID))

ON tblSubject.ID = tblCourse.SubjectID) ON tblTeacher.ID = tblCourse.TeacherID)

ON tblRoom.ID = tblCourse.RoomID) ON tblSchedule.ID = tblCourse.ScheduleID;"

Dim dt As DataTable = ExecuteQuery(query)

Return dt

End Function

Public Sub GetCourseByID(id As String)

Dim query As String = "SELECT \* FROM tblCourse WHERE ID = @ID;"

OpenConnection()

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@ID", id)

Dim reader As OleDbDataReader = cmd.ExecuteReader()

If reader.Read() Then

Me.ID = reader("ID").ToString()

subject.GetSubjectByID(reader("SubjectID").ToString())

teacher.GetTeacherByID(reader("TeacherID").ToString())

startDate = Convert.ToDateTime(reader("StartDate"))

room.GetRoomByID(Convert.ToInt16(reader("RoomID")))

scheduleID = Convert.ToInt32(reader("ScheduleID"))

endDate = If(IsDBNull(reader("EndDate")), Date.Now(), Convert.ToDateTime(reader("EndDate")))

statusID = Convert.ToInt32(reader("StatusID"))

End If

reader.Close()

CloseConnection()

End Sub

Public Sub AddCourse()

Dim query As String = "INSERT INTO tblCourse ( ID, SubjectID, TeacherID, StartDate, EndDate, RoomID, ScheduleID )

VALUES (@ID, @SubjectID, @TeacherID, @StartDate,@EndDate, @RoomID, @ScheduleID) "

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@ID", Convert.ToInt16(ID))

cmd.Parameters.AddWithValue("@SubjectID", Convert.ToInt16(subject.ID))

cmd.Parameters.AddWithValue("@TeacherID", Convert.ToInt16(teacher.TeacherID))

cmd.Parameters.AddWithValue("@StartDate", startDate)

cmd.Parameters.AddWithValue("@EndDate", endDate)

cmd.Parameters.AddWithValue("@RoomID", room.id)

cmd.Parameters.AddWithValue("@ScheduleID", scheduleID)

ExecuteNonQuery(cmd)

End Sub

Public Sub UpdateCourse()

Dim query As String = "UPDATE tblCourse

SET SubjectID = @subjectID, TeacherID = @TeacherID, StartDate = @StartDate, EndDate = @EndDate, RoomID = @RoomID, ScheduleID = @ScheduleID , StatusID = @StatusID

WHERE ID = @ID;"

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@subjectID", subject.ID)

cmd.Parameters.AddWithValue("@TeacherID", teacher.TeacherID)

cmd.Parameters.AddWithValue("@StartDate", startDate)

cmd.Parameters.AddWithValue("@EndDate", endDate)

cmd.Parameters.AddWithValue("@RoomID", room.id)

cmd.Parameters.AddWithValue("@ScheduleID", scheduleID)

cmd.Parameters.AddWithValue("@StatusID", statusID)

cmd.Parameters.AddWithValue("@ID", ID)

ExecuteNonQuery(cmd)

End Sub

Public Sub DeleteCourse()

Dim query As String = "DELETE FROM tblCourse WHERE ID = @ID;"

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@ID", ID)

ExecuteNonQuery(cmd)

End Sub

Public Function GetCurrentEnrollments() As Integer

Dim query As String = "SELECT CurrentEnrollment FROM tblCourse;"

OpenConnection()

Dim cmd As New OleDbCommand(query, GetConnection())

Dim count As Integer = Convert.ToInt32(cmd.ExecuteScalar())

Return count

End Function

'Get Dictionary to insert to Combobox

Public Function GetTimeList() As Dictionary(Of String, Integer)

Dim TimeList As New Dictionary(Of String, Integer)

Dim query As String = "SELECT tblSchedule.Schedule, tblSchedule.ID

FROM tblSchedule;"

OpenConnection()

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

Dim reader As OleDbDataReader = cmd.ExecuteReader()

While reader.Read()

TimeList.Add(reader("Schedule"), Integer.Parse(reader("ID")))

End While

reader.Close()

CloseConnection()

Return TimeList

End Function

Public Function GetRoomList() As Dictionary(Of String, Integer)

Dim RoomList As New Dictionary(Of String, Integer)

Dim query As String = "SELECT tblRoom.Room, tblRoom.ID

FROM tblRoom;"

OpenConnection()

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

Dim reader As OleDbDataReader = cmd.ExecuteReader()

While reader.Read()

RoomList.Add(reader("Room"), Integer.Parse(reader("ID")))

End While

reader.Close()

CloseConnection()

Return RoomList

End Function

Public Function GetClassStatusList() As Dictionary(Of String, Integer)

Dim StatusList As New Dictionary(Of String, Integer)

Dim query As String = "SELECT \* FROM tblCourseStatus;"

OpenConnection()

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

Dim reader As OleDbDataReader = cmd.ExecuteReader()

While reader.Read()

StatusList.Add(reader("Status"), Integer.Parse(reader("ID")))

End While

reader.Close()

CloseConnection()

Return StatusList

End Function

Public Function GetTeacherList() As Dictionary(Of String, String)

Dim TeacherList As New Dictionary(Of String, String)

Dim query As String = "SELECT tblTeacher.ID, tblTeacher.KhName FROM tblTeacher;"

OpenConnection()

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

Dim reader As OleDbDataReader = cmd.ExecuteReader()

While reader.Read()

TeacherList.Add(reader("KhName"), reader("ID"))

End While

reader.Close()

CloseConnection()

Return TeacherList

End Function

Public Function GetCourseList() As Dictionary(Of String, Integer)

Dim CourseList As New Dictionary(Of String, Integer)

Dim query As String = "SELECT tblSubject.Subject, tblSubject.ID FROM tblSubject;"

OpenConnection()

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

Dim reader As OleDbDataReader = cmd.ExecuteReader()

While reader.Read()

CourseList.Add(reader("Subject"), Integer.Parse(reader("ID")))

End While

reader.Close()

CloseConnection()

Return CourseList

End Function

Function GetSchedule() As String

Dim query As String = "SELECT Schedule FROM tblSchedule WHERE ID = @ScheduleID;"

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@ScheduleID", scheduleID)

OpenConnection()

Dim scheduleName As String = String.Empty

Dim reader As OleDbDataReader = cmd.ExecuteReader()

If reader.Read() Then

scheduleName = reader("Schedule").ToString()

End If

reader.Close()

CloseConnection()

Return scheduleName

End Function

Function GetStatus() As String

Dim query As String = "SELECT Status FROM tblCourseStatus WHERE ID = @StatusID;"

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@StatusID", statusID)

OpenConnection()

Dim statusName As String = String.Empty

Dim reader As OleDbDataReader = cmd.ExecuteReader()

If reader.Read() Then

statusName = reader("Status").ToString()

End If

reader.Close()

CloseConnection()

Return statusName

End Function

End Class

**20.Dashboard Class**

Imports System.Data.OleDb

Public Class Dashboard

Inherits ConnectionDB

Public totalStudent As Int16

Public completeStudent As Int16

Public totalCourse As Int16

Public completeCourse As Int16

Public totalTeacher As Int16

Public TotalSubject As Int16

Public CourseList As New Dictionary(Of String, Int16)

Public SubjectList As New Dictionary(Of String, Int16)

Public Sub New()

GetData()

End Sub

Public Sub GetData()

Dim query As String = "SELECT COUNT(\*) FROM tblRegister;"

totalStudent = Int16.Parse(ExecuteScalar(query))

query = "SELECT Count(\*) AS TotalStudent FROM tblRegister INNER JOIN tblCourse ON tblRegister.CourseID = tblCourse.ID GROUP BY tblCourse.StatusID HAVING (((tblCourse.StatusID)=3));"

Dim result = ExecuteScalar(query)

completeStudent = If(result IsNot Nothing, Convert.ToInt16(result), 0)

query = "SELECT COUNT(\*) FROM tblCourse;"

totalCourse = Int16.Parse(ExecuteScalar(query))

query = "SELECT COUNT(\*) FROM tblCourse WHERE StatusID = 3 ;"

completeCourse = Int16.Parse(ExecuteScalar(query))

query = "SELECT COUNT(\*) FROM tblTeacher;"

totalTeacher = Int16.Parse(ExecuteScalar(query))

query = "SELECT COUNT(\*) FROM tblSubject"

TotalSubject = Int16.Parse(ExecuteScalar(query))

query = "SELECT tblSubject.Subject, SUM(tblCourse.CurrentEnrollment) AS TotalStudent FROM tblCourse INNER JOIN tblSubject ON tblCourse.SubjectID = tblSubject.ID GROUP BY tblSubject.Subject;"

Dim cmd As New OleDbCommand(query, GetConnection())

OpenConnection()

Dim reader As OleDbDataReader = cmd.ExecuteReader()

While reader.Read()

CourseList.Add(reader("Subject").ToString(), Convert.ToInt16(reader("TotalStudent")))

End While

reader.Close()

cmd.CommandText = "SELECT tblSubject.Subject, (SELECT COUNT (\*) FROM tblCourse WHERE tblCourse.SubjectID = tblSubject.ID) AS TotalCourse FROM tblSubject"

reader = cmd.ExecuteReader()

While reader.Read()

SubjectList.Add(reader("Subject").ToString(), Convert.ToInt16(reader("TotalCourse")))

End While

reader.Close()

End Sub

'Courses DataTable

Public Function GetCourseDataTable() As DataTable

Dim query As String = "SELECT tblSubject.Subject, tblTeacher.KhName, (SELECT COUNT(\*) FROM tblRegister WHERE tblRegister.CourseID = tblCourse.ID) AS TotalStudent, tblCourse.StartDate, tblCourse.EndDate, tblCourseStatus.Status

FROM tblTeacher INNER JOIN (tblSubject INNER JOIN (tblCourseStatus INNER JOIN tblCourse ON (tblCourseStatus.ID = tblCourse.StatusID) AND (tblCourseStatus.ID = tblCourse.StatusID)) ON tblSubject.ID = tblCourse.SubjectID) ON tblTeacher.ID = tblCourse.TeacherID

WHERE (((tblCourse.StatusID)<=2));"

Dim dt = ExecuteQuery(query)

Return dt

End Function

Public Function GetNewRegister() As DataTable

Dim query As String = "SELECT \* FROM

(SELECT TOP 10 \*

FROM qtyRegister

ORDER BY ID DESC)

ORDER BY ID ASC;"

Dim dt = ExecuteQuery(query)

Return dt

End Function

Public Function CourseCoomingSoon() As DataTable

Dim query As String = "SELECT DCount( '\*','tblCourse','StartDate >= #' & Format([StartDate],'yyyy-mm-dd hh:nn : ss') & '#') AS [No], tblSubject.Subject, tblCourse.StartDate, tblCourse.CurrentEnrollment

From tblSubject INNER Join tblCourse On tblSubject.ID = tblCourse.SubjectID

Where (((tblCourse.StartDate) > Now()));"

Dim dt = ExecuteQuery(query)

Return dt

End Function

Public Function GetSubjectList() As DataTable

Dim query As String = " SELECT

tblSubject.ID,

tblSubject.Subject, COUNT(tblCourse.ID) AS TotalCourse,

SUM(IIf(tblCourse.StatusID = 3, 1, 0)) AS CompleteCourse,

SUM(tblCourse.CurrentEnrollment) AS SumOfCurrentEnrollment

FROM tblSubject INNER JOIN tblCourse ON tblSubject.ID = tblCourse.SubjectID

GROUP BY tblSubject.ID, tblSubject.Subject;"

Dim dt = ExecuteQuery(query)

Return dt

End Function

End Class

**21.Payment Class**

Imports System.Data.OleDb

Public Class Payment

Inherits ConnectionDB

Public paymentID As String

Public registerID As String

Public amount As Decimal

Public unpaid As Decimal

Public statusID As String

'Method

Public Sub NewPayment(registerID As String, amount As Decimal, unpaid As Decimal)

Dim statusID As String = If(unpaid > 0, "1", "2")

Dim query As String = "INSERT INTO tblPayment(RegisterID, Amount, Unpaid ,StatusID) VALUES(@registerID, @amount, @unpaid ,@statusID);"

OpenConnection()

Dim cmd As New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@registerID", registerID)

cmd.Parameters.AddWithValue("@amount", amount)

cmd.Parameters.AddWithValue("@unpaid", unpaid)

cmd.Parameters.AddWithValue("@statusID", statusID)

cmd.ExecuteNonQuery()

cmd.CommandText = "SELECT @@IDENTITY;"

Dim id As String = cmd.ExecuteScalar().ToString()

paymentID = id

Me.amount = amount

Me.unpaid = unpaid

Me.registerID = registerID

Me.statusID = statusID

cmd.Dispose()

CloseConnection()

End Sub

Public Sub GetPaymentByID(registerID As String)

Dim query As String = "SELECT \* FROM tblPayment WHERE RegisterID = @id;"

OpenConnection()

Dim cmd As New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@id", registerID)

Dim reader As OleDbDataReader = cmd.ExecuteReader()

If reader.Read() Then

paymentID = reader("ID")

registerID = reader("RegisterID")

amount = Convert.ToDecimal(reader("Amount"))

unpaid = Convert.ToDecimal(reader("Unpaid"))

statusID = reader("StatusID")

End If

reader.Close()

cmd.Dispose()

CloseConnection()

End Sub

Public Sub UpdatePayment(registerID As String)

Dim query As String = "UPDATE tblePayment SET Unpaid = @unpaid WHERE RegisterID = @id;"

Dim cmd As New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@id", registerID)

ExecuteNonQuery(cmd)

UpdateStatus(registerID)

End Sub

Private Sub UpdateStatus(registerID As String)

Dim query As String = "UPDATE tblPayment SET StatusID = IIf(Unpaid > 0, 2, 1) WHERE REgisterID = @id;"

Dim cmd As New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@id", registerID)

ExecuteNonQuery(cmd)

End Sub

End Class

**22.Register Class**

Imports System.Data.OleDb

Imports System.Web.UI

Public Class Register

Inherits ConnectionDB

Public registerID As String

Public course As Course

Public student As Student

Public discount As Decimal

Public payment As Payment

'Method to register a student for a subject

Public Function GetRegisterData() As DataTable

Dim query As String = "SELECT tblCourse.ID, tblSubject.Subject, tblTeacher.EngName, tblCourse.StartDate, tblSchedule.Schedule, tblRoom.Room, (SELECT COUNT(\*) FROM tblRegister WHERE tblCourse.ID = tblRegister.CourseID) AS TotalStudent

FROM tblTeacher INNER JOIN (tblSchedule INNER JOIN (tblRoom INNER JOIN (tblSubject INNER JOIN (tblCourseStatus INNER JOIN tblCourse ON tblCourseStatus.ID = tblCourse.StatusID) ON tblSubject.ID = tblCourse.SubjectID) ON tblRoom.ID = tblCourse.RoomID) ON tblSchedule.ID = tblCourse.ScheduleID) ON tblTeacher.ID = tblCourse.TeacherID

WHERE (((tblCourse.StatusID)<=2))

ORDER BY tblCourse.ID;"

Dim dt As DataTable = ExecuteQuery(query)

Return dt

End Function

Public Sub New()

' Constructor for Register class

MyBase.New()

course = New Course()

student = New Student()

payment = New Payment()

End Sub

'Register a student for a class

Public Sub RegisterStudent()

Dim query As String = "INSERT INTO tblRegister (CourseID, StudentID, Discount) VALUES (@CourseID, @StudentID, @Discount);"

OpenConnection()

Dim cmd As New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@CourseID", course.ID)

cmd.Parameters.AddWithValue("@StudentID", student.id)

cmd.Parameters.AddWithValue("@Discount", discount)

cmd.ExecuteNonQuery()

cmd.CommandText = "SELECT @@IDENTITY;"

Dim id As String = cmd.ExecuteScalar().ToString()

query = "UPDATE tblCourse SET CurrentEnrollment = CurrentEnrollment + 1 WHERE ID = @id"

cmd.CommandText = query

cmd.Parameters.AddWithValue("@id", course.ID)

ExecuteNonQuery(cmd)

GetRegisterByID(id)

End Sub

'Get register by ID

Public Sub GetRegisterByID(id As String)

Dim query As String = "SELECT \* FROM tblRegister WHERE ID = @RegisterID;"

OpenConnection()

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@RegisterID", id)

Dim reader As OleDbDataReader = cmd.ExecuteReader()

If reader.Read() Then

registerID = reader("ID").ToString()

course.GetCourseByID(reader("CourseID").ToString())

student.GetStudentByID(reader("StudentID").ToString())

discount = Convert.ToByte(reader("Discount"))

payment.GetPaymentByID(registerID)

End If

reader.Close()

CloseConnection()

End Sub

'Update register information

Public Sub UpdateRegister(id As String)

Dim query As String = "UPDATE tblRegister SET CourseID = @CourseID, StudentID = @StudentID, Discount = @Discount WHERE RegisterID = @RegisterID;"

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@CourseID", course.ID)

cmd.Parameters.AddWithValue("@StudentID", student.id)

cmd.Parameters.AddWithValue("@Discount", discount)

cmd.Parameters.AddWithValue("@RegisterID", id)

ExecuteNonQuery(cmd)

End Sub

'Delete a register entry by ID

Public Sub DeleteRegister(id As String)

Dim query As String = "DELETE FROM tblRegister WHERE RegisterID = @RegisterID;"

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@RegisterID", id)

ExecuteNonQuery(cmd)

query = "UPDATE tblCourse SET CurrentEnrollment = CurrentEnrollment - 1 WHERE ID IN (SELECT ClassID FROM tblRegister WHERE ID = @id"

cmd.CommandText = query

cmd.Parameters.AddWithValue("@id", id)

ExecuteNonQuery(cmd)

End Sub

'Insert DATA TO combobox

Public Function GetSubjectList() As Dictionary(Of String, Integer)

Dim subjectList As New Dictionary(Of String, Integer)

Dim query As String = "SELECT DISTINCT tblSubject.Subject, tblSubject.ID

FROM tblSubject INNER JOIN tblCourse ON tblSubject.ID = tblCourse.SubjectID

GROUP BY tblSubject.Subject, tblSubject.ID, tblCourse.StatusID, tblCourse.SubjectID

HAVING (((tblCourse.StatusID)<=2));"

OpenConnection()

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

Dim reader As OleDbDataReader = cmd.ExecuteReader()

While reader.Read()

subjectList.Add(reader("Subject"), Integer.Parse(reader("ID")))

End While

subjectList.Add("ជ្រើសរើសវគ្គសិក្សា", -1)

reader.Close()

CloseConnection()

Return subjectList

End Function

Public Function GetTimeList(subjectID As String) As Dictionary(Of String, Integer)

Dim TimeList As New Dictionary(Of String, Integer)

Dim query As String = "SELECT tblSchedule.Schedule, tblCourse.ID

FROM tblSchedule INNER JOIN tblCourse ON tblSchedule.ID = tblCourse.ScheduleID

WHERE (((tblCourse.SubjectID)=@ID) AND ((tblCourse.StatusID)<=2))

ORDER BY tblSchedule.Schedule;"

OpenConnection()

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@ID", subjectID)

Dim reader As OleDbDataReader = cmd.ExecuteReader()

While reader.Read()

TimeList.Add(reader("Schedule"), Integer.Parse(reader("ID")))

End While

TimeList.Add("ជ្រើសរើសម៉ោង", -1)

reader.Close()

CloseConnection()

Return TimeList

End Function

End Class

**23.Report Class**

Imports System.Data.OleDb

Public Class Report

Inherits ConnectionDB

Public Function GetTeacherReport() As DataTable

Dim query As String = "SELECT tblTeacher.ID, tblTeacher.KhName, tblTeacher.engName, tblTeacher.Address, tblTeacher.Phone, tblTeacher.Email, tblTeacher.StartDate, tblTeacher.IsActive FROM tblTeacher;"

OpenConnection()

Dim cmd As New OleDbCommand(query, GetConnection())

Dim reader As OleDbDataReader = cmd.ExecuteReader()

Dim dt As New DataTable()

dt.Columns.Add("ID", GetType(String))

dt.Columns.Add("KhName", GetType(String))

dt.Columns.Add("engName", GetType(String))

dt.Columns.Add("Address", GetType(String))

dt.Columns.Add("Phone", GetType(String))

dt.Columns.Add("Email", GetType(String))

dt.Columns.Add("StartDate", GetType(String)) ' Change to String for formatting

dt.Columns.Add("IsActive", GetType(String))

While reader.Read()

Dim formattedDate As String = Convert.ToDateTime(reader("StartDate")).ToString("dd/MM/yyyy") ' Example format

Dim formattedActive As String = If(reader("IsActive").ToString().ToLower() = "true", "សកម្ម", "អសកម្ម")

dt.Rows.Add(reader("ID"), reader("KhName"), reader("EngName"), reader("Address"),

reader("Phone"), reader("Email"), formattedDate, formattedActive)

End While

Return dt

End Function

Public Function GetStudentReport() As DataTable

Dim query As String = "SELECT tblStudent.ID, tblStudent.KhName, tblStudent.EngName, tblStudent.Gender, tblStudent.DateOfBirth, tblStudent.Address, tblStudent.Phone, tblSubject.Subject

FROM ((tblStudent INNER JOIN tblRegister ON tblStudent.ID = tblRegister.StudentID) INNER JOIN tblCourse ON tblRegister.CourseID = tblCourse.ID) INNER JOIN tblSubject ON tblCourse.SubjectID = tblSubject.ID;"

OpenConnection()

Dim cmd As New OleDbCommand(query, GetConnection())

Dim reader As OleDbDataReader = cmd.ExecuteReader()

Dim dt As New DataTable()

dt.Columns.Add("ID", GetType(Integer))

dt.Columns.Add("KhName", GetType(String))

dt.Columns.Add("EngName", GetType(String))

dt.Columns.Add("Gender", GetType(String))

dt.Columns.Add("Address", GetType(String))

dt.Columns.Add("Phone", GetType(String))

dt.Columns.Add("DateOfBirth", GetType(String)) ' Change to String for formatting

dt.Columns.Add("Subject", GetType(String))

While reader.Read()

Dim formattedDate As String = Convert.ToDateTime(reader("DateOfBirth")).ToString("dd/MM/yyyy") ' Example format

dt.Rows.Add(reader("ID"), reader("KhName"), reader("EngName"), reader("Gender"), reader("Address"),

reader("Phone"), formattedDate, reader("Subject"))

End While

Return dt

End Function

Public Function GetCourseReport() As DataTable

Dim query As String = "SELECT tblCourse.ID, tblSubject.Subject, tblTeacher.KhName, tblSchedule.Schedule, tblRoom.Room, tblCourse.StartDate,

tblCourse.EndDate, tblCourse.CurrentEnrollment, tblCourseStatus.Status

FROM tblCourseStatus INNER JOIN (tblTeacher INNER JOIN (tblSchedule

INNER JOIN (tblRoom INNER JOIN (tblSubject INNER JOIN tblCourse ON tblSubject.ID = tblCourse.[SubjectID])

ON tblRoom.ID = tblCourse.RoomID) ON tblSchedule.ID = tblCourse.ScheduleID) ON tblTeacher.ID = tblCourse.TeacherID)

ON (tblCourseStatus.ID = tblCourse.StatusID) AND (tblCourseStatus.ID = tblCourse.StatusID);"

OpenConnection()

Dim cmd As New OleDbCommand(query, GetConnection())

Dim reader As OleDbDataReader = cmd.ExecuteReader()

Dim dt As New DataTable()

dt.Columns.Add("ID", GetType(Integer))

dt.Columns.Add("Subject", GetType(String))

dt.Columns.Add("KhName", GetType(String))

dt.Columns.Add("Schedule", GetType(String))

dt.Columns.Add("Room", GetType(String))

dt.Columns.Add("StartDate", GetType(String))

dt.Columns.Add("EndDate", GetType(String)) ' Change to String for formatting

dt.Columns.Add("CurrentEnrollment", GetType(Integer))

dt.Columns.Add("Status", GetType(String))

While reader.Read()

Dim startDate As String = Convert.ToDateTime(reader("StartDate")).ToString("dd/MM/yyyy") ' Example format

Dim EndDate As String = Convert.ToDateTime(reader("EndDate")).ToString("dd/MM/yyyy")

dt.Rows.Add(reader(0), reader(1), reader(2), reader(3), reader(4),

startDate, EndDate, reader(7), reader(8))

End While

Return dt

End Function

Public Function GetSubjectReport() As DataTable

Dim query As String = "SELECT tblSubject.ID, tblSubject.Subject, tblSubject.Description,

tblSubject.CreditHours, tblSubject.BasePrice, (SELECT COUNT(\*)

FROM tblCourse WHERE tblCourse.SubjectID = tblSubject.ID) AS TotalCourse,

(SELECT SUM(IIf(IsNull(CurrentEnrollment), 0, CurrentEnrollment))

FROM tblCourse WHERE tblCourse.SubjectID = tblSubject.ID) AS TotalStudent

FROM tblSubject;"

OpenConnection()

Dim cmd As New OleDbCommand(query, GetConnection())

Dim reader As OleDbDataReader = cmd.ExecuteReader()

Dim dt As New DataTable()

dt.Columns.Add("ID", GetType(Integer))

dt.Columns.Add("Subject", GetType(String))

dt.Columns.Add("Description", GetType(String))

dt.Columns.Add("CreditHours", GetType(String))

dt.Columns.Add("BasePrice", GetType(String))

dt.Columns.Add("TotalCourse", GetType(String))

dt.Columns.Add("TotalStudent", GetType(String)) ' Change to String for formatting

While reader.Read()

Dim totalStudent = If(reader("TotalStudent").ToString() = String.Empty, 0, Convert.ToInt16(reader("TotalStudent")))

dt.Rows.Add(reader(0), reader(1), reader(2), reader(3), reader(4),

reader(5), totalStudent)

End While

Return dt

End Function

Public Function GetPaymentReport() As DataTable

Dim query As String = "SELECT tblStudent.ID, tblStudent.KhName, tblSubject.Subject, tblSubject.BasePrice,

tblRegister.Discount,tblPayment.Amount,tblPayment.Unpaid,tblPaymentStatus.Status FROM tblStudent INNER JOIN ((tblSubject INNER JOIN tblCourse ON tblSubject.ID = tblCourse.SubjectID) INNER JOIN ((tblPaymentStatus INNER JOIN tblPayment ON tblPaymentStatus.ID = tblPayment.StatusID) INNER JOIN tblRegister ON tblPayment.RegisterID = tblRegister.ID) ON tblCourse.ID = tblRegister.CourseID) ON tblStudent.ID = tblRegister.StudentID;"

OpenConnection()

Dim cmd As New OleDbCommand(query, GetConnection())

Dim reader As OleDbDataReader = cmd.ExecuteReader()

Dim dt As New DataTable()

dt.Columns.Add("ID", GetType(String))

dt.Columns.Add("KhName", GetType(String))

dt.Columns.Add("Subject", GetType(String))

dt.Columns.Add("BasePrice", GetType(Decimal))

dt.Columns.Add("Discount", GetType(Decimal))

dt.Columns.Add("Amount", GetType(Decimal))

dt.Columns.Add("Unpaid", GetType(Decimal)) ' Change to String for formatting

dt.Columns.Add("Status", GetType(String))

While reader.Read()

dt.Rows.Add(reader(0), reader(1), reader(2), Convert.ToDecimal(reader(3)), Convert.ToDecimal(reader(4)), Convert.ToDecimal(reader(5)), Convert.ToDecimal(reader(6)), reader(7))

End While

Return dt

End Function

End Class

**24.Room Class**

Imports System.Data.OleDb

Imports System.Diagnostics.Eventing.Reader

Public Class Room

Inherits ConnectionDB

Public id As Integer

Public room As String

Public Function GetRoomList() As DataTable

Dim query As String = "SELECT \* FROM tblRoom;"

Dim dt As DataTable = ExecuteQuery(query)

Return dt

End Function

Public Sub GetRoomByID(id As Integer)

Dim query As String = "SELECT \* FROM tblRoom WHERE ID = " & id & ";"

OpenConnection()

Dim cmd As New OleDbCommand(query, GetConnection())

Dim reader As OleDbDataReader = cmd.ExecuteReader()

If reader.Read() Then

Me.id = reader(0)

Me.room = reader(1)

End If

End Sub

End Class

**25.Student Class**

Imports System.ComponentModel

Imports System.Data.OleDb

Public Class Student

Inherits ConnectionDB

Public id As String

Public khName As String

Public engName As String

Public DateOfBirth As Date

Public gender As String

Public phone As String

Public address As String

Public picture As String

Public Sub New()

' Constructor for Student class

MyBase.New()

End Sub

Public Function GetStudentData() As DataTable

Dim query As String = "SELECT tblStudent.ID, tblStudent.KhName, tblStudent.EngName, tblStudent.Gender, tblStudent.DateOfBirth, tblStudent.Address , tblStudent.Phone FROM tblStudent;"

Dim dt As DataTable = ExecuteQuery(query)

Return dt

End Function

Public Sub AddStudent(khName As String, EngName As String, Gender As String, DOB As Date, Address As String, Phone As String, Picture As String)

Dim query As String = "INSERT INTO tblStudent(KhName , EngName ,Gender , DateOfBirth , Address , Phone ,Picture )

VALUES (@KhName, @EngName, @Gender, @DateOfBirth, @Address, @Phone, @Picture)"

OpenConnection()

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@KhName", khName)

cmd.Parameters.AddWithValue("@EngName", EngName)

cmd.Parameters.AddWithValue("@Gender", Gender)

cmd.Parameters.AddWithValue("@DateOfBirth", DOB)

cmd.Parameters.AddWithValue("@Address", Address)

cmd.Parameters.AddWithValue("@Phone", Phone)

cmd.Parameters.AddWithValue("@Picter", Picture)

cmd.ExecuteNonQuery()

cmd.CommandText = "SELECT @@IDENTITY;"

Dim id As String = cmd.ExecuteScalar().ToString()

GetStudentByID(Convert.ToInt32(id))

End Sub

Public Sub UpdateStudent(studentID As Integer, khName As String, EngName As String, Gender As String, DOB As Date, Address As String, Phone As String, Picture As String)

Dim query As String = "UPDATE tblStudent

SET KhName = @KhName, EngName = @EngName, Gender = @Gender, DateOfBirth = @DateOfBirth,

Address = @Address, Phone = @Phone, Picture = @Picture

WHERE ID = @StudentID"

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@KhName", khName)

cmd.Parameters.AddWithValue("@EngName", EngName)

cmd.Parameters.AddWithValue("@Gender", Gender)

cmd.Parameters.AddWithValue("@DateOfBirth", DOB)

cmd.Parameters.AddWithValue("@Address", Address)

cmd.Parameters.AddWithValue("@Phone", Phone)

cmd.Parameters.AddWithValue("@Picture", Picture)

cmd.Parameters.AddWithValue("@StudentID", studentID) ' Ensure you reference the correct ID field

ExecuteNonQuery(cmd)

End Sub

Public Sub DeleteStudent()

Dim query As String = "DELETE FROM tblStudent WHERE ID = @ID;"

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@ID", Me.id)

ExecuteNonQuery(cmd)

MessageBox.Show("Student with ID: " & Me.id & " has been deleted successfully.", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information)

End Sub

Public Sub GetStudentByID(studentID As Integer)

Dim query As String = "SELECT \* FROM tblStudent WHERE ID = @StudentID"

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@StudentID", studentID)

OpenConnection()

Dim reader As OleDbDataReader = cmd.ExecuteReader()

While reader.Read()

Me.id = reader("ID").ToString()

Me.khName = reader("KhName").ToString()

Me.engName = reader("EngName").ToString()

Me.DateOfBirth = reader("DateOfBirth").ToString()

Me.gender = reader("Gender").ToString()

Me.phone = reader("Phone").ToString()

Me.address = reader("Address").ToString()

Me.picture = reader("Picture").ToString()

End While

End Sub

Public Function GetAddressData() As ArrayList

Dim addressList As New ArrayList()

Dim query As String = "SELECT \* FROM tblProvince;"

OpenConnection()

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

Dim reader As OleDbDataReader = cmd.ExecuteReader()

While reader.Read()

addressList.Add(reader(0).ToString)

End While

reader.Close()

cmd.Dispose()

CloseConnection()

Return addressList

End Function

End Class

**26.Subject Clas**

Imports System.Data.OleDb

Public Class Subject

Inherits ConnectionDB

Public Sub New()

' Constructor for Course class

MyBase.New()

End Sub

Public ID As String

Public Subject As String

Public Description As String

Public CreditHours As Integer

Public basePrice As Decimal

Public Function GetSubjectData() As DataTable

Dim query As String = "SELECT

tblSubject.ID,

tblSubject.Subject,

tblSubject.Description,

tblSubject.CreditHours,

tblSubject.BasePrice,

(SELECT COUNT(\*) FROM tblCourse WHERE tblCourse.SubjectID = tblSubject.ID) AS AllCourse,

(SELECT COUNT(\*) FROM tblCourse WHERE StatusID = 3 AND tblCourse.SubjectID = tblSubject.ID) AS CompleteClass

FROM tblSubject

GROUP BY

tblSubject.ID,

tblSubject.Subject,

tblSubject.Description,

tblSubject.CreditHours,

tblSubject.BasePrice;"

Dim dt As DataTable = ExecuteQuery(query)

Return dt

End Function

Public Sub GetSubjectByID(courseID As String)

Dim query As String = "SELECT \* FROM tblSubject WHERE ID = @ID"

OpenConnection()

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@ID", courseID)

Dim reader As OleDbDataReader = cmd.ExecuteReader()

While reader.Read()

Me.ID = reader("ID").ToString()

Me.Subject = reader("Subject").ToString()

Me.Description = reader("Description").ToString()

Me.CreditHours = Convert.ToInt32(reader("CreditHours"))

Me.basePrice = Convert.ToDecimal(reader("BasePrice"))

End While

reader.Close()

CloseConnection()

End Sub

Public Sub GetSubjectByName(subject As String)

Dim query As String = "SELECT \* FROM tblSubject WHERE Subject = @subject"

OpenConnection()

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@subject", Me.Subject)

Dim reader As OleDbDataReader = cmd.ExecuteReader()

While reader.Read()

Me.ID = reader("ID").ToString()

Me.Subject = reader("Subject").ToString()

Me.Description = reader("Description").ToString()

Me.CreditHours = Convert.ToInt32(reader("CreditHours"))

Me.basePrice = Convert.ToDecimal(reader("BasePrice"))

End While

reader.Close()

CloseConnection()

End Sub

Public Sub AddSubject(subject As String, description As String, creditHours As Integer, basePrice As Decimal)

Dim query As String = "INSERT INTO tblSubject (Subject, Description, CreditHours, BasePrice) VALUES (@subject, @Description, @CreditHours, @BasePrice)"

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@subject", subject)

cmd.Parameters.AddWithValue("@Description", description)

cmd.Parameters.AddWithValue("@CreditHours", creditHours)

cmd.Parameters.AddWithValue("@BasePrice", basePrice)

ExecuteNonQuery(cmd)

End Sub

Public Sub DeleteSubject()

Dim query As String = "DELETE FROM tblSubject WHERE ID = @ID"

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@ID", Me.ID)

ExecuteNonQuery(cmd)

End Sub

Public Sub UpdateSubject()

Dim query As String = "UPDATE tblSubject SET Subject = @subject, Description = @Description, CreditHours = @CreditHours, BasePrice = @BasePrice WHERE ID = @ID"

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@subject", Me.Subject)

cmd.Parameters.AddWithValue("@Description", Me.Description)

cmd.Parameters.AddWithValue("@CreditHours", Me.CreditHours)

cmd.Parameters.AddWithValue("@BasePrice", Me.basePrice)

cmd.Parameters.AddWithValue("@ID", Me.ID)

ExecuteNonQuery(cmd)

End Sub

End Class

**27.Teacher Class**

Imports System.Data.OleDb

Public Class Teacher

Inherits ConnectionDB

Public Sub New()

' Constructor for Teacher class

MyBase.New()

End Sub

Public TeacherID As String

Public KhName As String

Public EngName As String

Public Gender As String

Public DoB As Date

Public Address As String

Public Phone As String

Public Email As String

Public Picture As String

Public Function GetTeacherData() As DataTable

Dim query As String = "SELECT tblTeacher.ID, tblTeacher.KhName, tblTeacher.EngName, tblTeacher.DOB, tblTeacher.Address, tblTeacher.Email, tblTeacher.Phone FROM tblTeacher;"

Dim dt As DataTable = ExecuteQuery(query)

Return dt

End Function

Public Sub AddTeacher()

Dim query As String = "INSERT INTO tblTeacher(khName , EngName ,Gender , DOB , Address , Email , Phone ,Picture ,IsActive)

VALUES (@KhName, @EngName, @Gender, @DoB, @Address, @Email, @Phone, @Picture, 1)"

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@KhName", Me.KhName)

cmd.Parameters.AddWithValue("@EngName", Me.EngName)

cmd.Parameters.AddWithValue("@Gender", Me.Gender)

cmd.Parameters.AddWithValue("@DoB", Me.DoB)

cmd.Parameters.AddWithValue("@Address", Me.Address)

cmd.Parameters.AddWithValue("@Email", Me.Email)

cmd.Parameters.AddWithValue("@Phone", Me.Phone)

cmd.Parameters.AddWithValue("@Picter", Me.Picture)

ExecuteNonQuery(cmd)

End Sub

Public Sub UpdateTeacher()

Dim query As String = "UPDATE tblTeacher SET KhName = @KhName, EngName = @engName, Gender = @Gender, DOB = @DOB, Address = @Address, Email = @Email, Phone = @Phone, Picture = @Picture WHERE ID = @ID"

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@khName", KhName)

cmd.Parameters.AddWithValue("@EngName", EngName)

cmd.Parameters.AddWithValue("@Gender", Gender)

cmd.Parameters.AddWithValue("@DOB", DoB)

cmd.Parameters.AddWithValue("@Address", Me.Address)

cmd.Parameters.AddWithValue("@Email", Email)

cmd.Parameters.AddWithValue("@Phone", Phone)

cmd.Parameters.AddWithValue("@Picture", Picture)

cmd.Parameters.AddWithValue("@ID", TeacherID)

ExecuteNonQuery(cmd)

End Sub

Public Sub DeleteTeacher(teacherID As String)

Dim query As String = "DELETE FROM tblTeacher WHERE ID = @ID;"

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@ID", teacherID)

ExecuteNonQuery(cmd)

End Sub

Public Sub GetTeacherByID(ID As String)

Dim query As String = "SELECT \* FROM tblTeacher WHERE ID = @ID;"

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@ID", ID)

OpenConnection()

Dim reader As OleDbDataReader = cmd.ExecuteReader()

While reader.Read()

Me.TeacherID = reader("ID").ToString()

Me.KhName = reader("KhName").ToString()

Me.EngName = reader("EngName").ToString()

Me.Address = reader("Address").ToString()

Me.DoB = Convert.ToDateTime(reader("DOB"))

Me.Gender = reader("Gender").ToString()

Me.Phone = reader("Phone").ToString()

Me.Email = reader("Email").ToString()

If Not IsDBNull(reader("Picture")) Then

Me.Picture = reader("Picture").ToString()

Else

Me.Picture = String.Empty

End If

End While

reader.Close()

CloseConnection()

End Sub

Public Sub GetTeacherByName(name As String)

Dim query As String = "SELECT \* FROM tblTeacher WHERE KhName = @khName;"

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

cmd.Parameters.AddWithValue("@khName", name)

OpenConnection()

Dim reader As OleDbDataReader = cmd.ExecuteReader()

While reader.Read()

Me.TeacherID = reader("ID").ToString()

Me.KhName = reader("KhName").ToString()

Me.EngName = reader("EngName").ToString()

Me.Address = reader("Address").ToString()

Me.DoB = Convert.ToDateTime(reader("DOB"))

Me.Gender = reader("Gender").ToString()

Me.Phone = reader("Phone").ToString()

Me.Email = reader("Email").ToString()

If Not IsDBNull(reader("Picture")) Then

Me.Picture = reader("Picture").ToString()

Else

Me.Picture = String.Empty

End If

End While

reader.Close()

CloseConnection()

End Sub

Public Function GetAddressData() As ArrayList

Dim addressList As New ArrayList()

Dim query As String = "SELECT \* FROM tblProvince;"

Dim cmd As OleDbCommand = New OleDbCommand(query, GetConnection())

OpenConnection()

Dim reader As OleDbDataReader = cmd.ExecuteReader()

While reader.Read()

addressList.Add(reader(0).ToString)

End While

reader.Close()

CloseConnection()

Return addressList

End Function

End Class

**28.ThemeColor**

Public Class ThemeColor

Public Shared menuBackColor As Color = Color.FromArgb(5, 43, 80)

Public Shared activeColor As Color = Color.FromArgb(17, 139, 254)

Public Shared DashboardColor As Color = Color.FromArgb(63, 114, 236)

Public Shared CourseColor As Color = Color.FromArgb(143, 135, 241)

'Public Shared btnStudenBackColor As Color =

Public Shared Function ChangeColorBrightness(color As Color, correctionFactor As Double) As Color

Dim red As Double = color.R

Dim green As Double = color.G

Dim blue As Double = color.B

If correctionFactor < 0 Then

correctionFactor += 1

red \*= correctionFactor

green \*= correctionFactor

blue \*= correctionFactor

Else

red = (255 - red) \* correctionFactor + red

green = (255 - green) \* correctionFactor + green

blue = (255 - blue) \* correctionFactor + blue

End If

Return Color.FromArgb(color.A, Math.Round(red), Math.Round(green), Math.Round(blue))

End Function

End Class