



SEMESTER 2
SESSION 2022/2023

SECP3106-01
APPLICATION DEVELOPMENT (WBL)

Individual Test - FYP Assignment

Case Study: FYP1 System

Project Report

NAME	MATRIC NUMBER
LOW JUNYI	A20EC0071

LECTURER: DR. HAZA NUZLY BIN ABDULL HAMED

Table Of Contents

1.0 Introduction	3
2.0 Software, language, tools, API, template used	4
3.0 Development steps	6
3.1 Entity Relationship Diagram (ERD) & Database Design	6
3.2 Start new project using .NET web application (C#) with MVC5	8
3.3 Add bootstrap template into asp.net project	9
3.3.1 Change the layout and add in datatable features	9
3.4 Add in Entity Data Model and all related controllers	11
3.5 Add Models and Views	12
3.5.1 Login & Logout	12
3.5.2 Forgot Password Features	14
3.6 Academic Program Module	16
3.6.1 Create Role-Based Views	17
3.7 Users Module	18
3.7.1 Create Role-Based Views	18
3.8 PIC in Proposal Module	25
3.8.1 Add a folder for Supervisor Agreement	25
3.8.2 Create Role-Based Views	26
3.9 Proposal Module	33
3.9.1 Add a folder for All Students Proposals	33
3.9.2 Create Role-Based Views	34
3.10 Customise the Home Index, Navigation Sidebar and All Modules	41
4.0 System interface for all users	42
4.2.1 Admin	45
Home Page	45
Menu 1: Academic Program	45
Menu 2: All Lecturers	49
4.2.2 Student	52
Home Page	52
Menu 1: PIC in proposals	52
Menu 2: Submit Proposals	55
4.2.3 Committee	59
Home Page	59
Menu 1: All Users	59
Menu 2: PIC in Proposals	62
Menu 3: All Proposals	64
4.2.4 Lecturers (Supervisors/Evaluators)	67

Home Page	67
Menu 1: Students	67
Menu 2: All Proposals	70
5.0 Localhost Installation Steps	74
6.0 User credentials	81
7.0 Documentation: SDD	82

1.0 Introduction



This report is concerning a Final Year Project Proposal (FYP1) System of UTM developed using the .NET web application (C#) platform. In this individual test, I am going to develop a FYP1 system which is used for submitting and evaluating the proposal of FYP. I am required to build the FYP system with five user types which are admin, student, supervisor, evaluator and committee. In this system, I separate the types into admin, student, lecturers (supervisor, evaluator) and committee. The committee will be chosen from the list of lecturers. Once chosen as committee, the role of that user will be changed from lecturer into committee which can manage all students and proposals whereas the supervisor and evaluator can only view their own students' proposals.

Flows:

At the very beginning, the admin will first manage the academic program and lecturer list, and assign a committee based on lecturer lists and programs. Next, the committee will assign domain (research / development) to the lecturer lists created by admin. Students register personal details on the system including choosing project type. After registering successfully, students login to the system and select a supervisor (SV), waiting for approval from the committee. Once students choose the preferred supervisor, status of choosing supervisor will be pending. Selected SV will be able to view at which students select themselves and upload agreements. The committee will approve SV upload agreements. Students will be able to view agreements once the committee approves it. Next, the committee will assign evaluator 1 and evaluator 2 to the students and the person in charge will be three different people but they have the same domain and academic program with the students. Students are allowed to upload proposals once supervisor is approved and evaluators 1 and 2 will be able to download all of the uploaded assigned proposals. Evaluator 1 and 2 will leave comments, give evaluation results whether is accepted, accepted with conditions or rejected after SV left comments. Students can resubmit proposals if the status is accepted with conditions or rejected. Supervisors and evaluators can view all related proposals as long as they are the PIC in each of the proposals. The committee can manage all proposals.

2.0 Software, language, tools, API, template used

Software

Software used in this AD Test are **Microsoft SQL Server Management Studio (SSMS)** and **Microsoft Visual Studio 2022**. Microsoft SQL Server Management Studio (SSMS) is where we used to create a database, manage and store the tables. Microsoft Visual Studio is where we used to create a project with a platform of .NET web application (C#) along with the MVC5 framework. Then connect it to the database created in SSMS. All of the controllers, views and all related code are edited using this software.

Language

Major language used in this system including Hypertext Markup Language (HTML), C# (C-Sharp) and Cascading Style Sheet (CSS). HTML and CSS is used for the front end development. C# is used for the back end development of a system. Javascript and jQuery are used for front end development too, such as scroll to top, sorting, pagination and live search using table rows. Language-Integrated Query (LINQ) in C# will work with data from SQL Server to retrieve or manage data.

Tools

Draw.io is used for the drawing of ERD for this system before creating a database in SSMS. Enterprise Architecture will be used in preparing the diagrams in SDD. Microsoft Visual Studio 2022 is used to edit the code. It consists of adding controller features which will automatically create views of index, create, edit, details and delete which fulfils the CRUD operations will make things easier.

API and Templates

The Bootstrap template which I used in this AD Test is taken from [BootstrapMade](#). It is a website that provides many free bootstrap HTML templates which I can use to apply as layout in my project. The name of the template I used is an admin and dashboard template named NiceAdmin. Below is the URL of my template.

<https://bootstrapmade.com/nice-admin-bootstrap-admin-html-template/>

Google Fonts

```
<link href="https://fonts.gstatic.com" rel="preconnect">
<link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i|Nunito:300,300i,400,400i,600,600i,700,700i|Poppins:300,300i,400,400i,500,500i,600,600i,700,700i" rel="stylesheet">
```

Below shows the important element for datatable which shows the pagination, search and sort.

```
<script src="~/Content/assets/vendor/simple-datatables/simple-datatables.js"></script>
```

By using the template, we have to also include the paths of Favicons, Google Fonts, Vendor CSS files, Vendor JS files and Template Main JS File so that all features of the templates will be shown and fully utilised.

3.0 Development steps

3.1 Entity Relationship Diagram (ERD) & Database Design

ERD

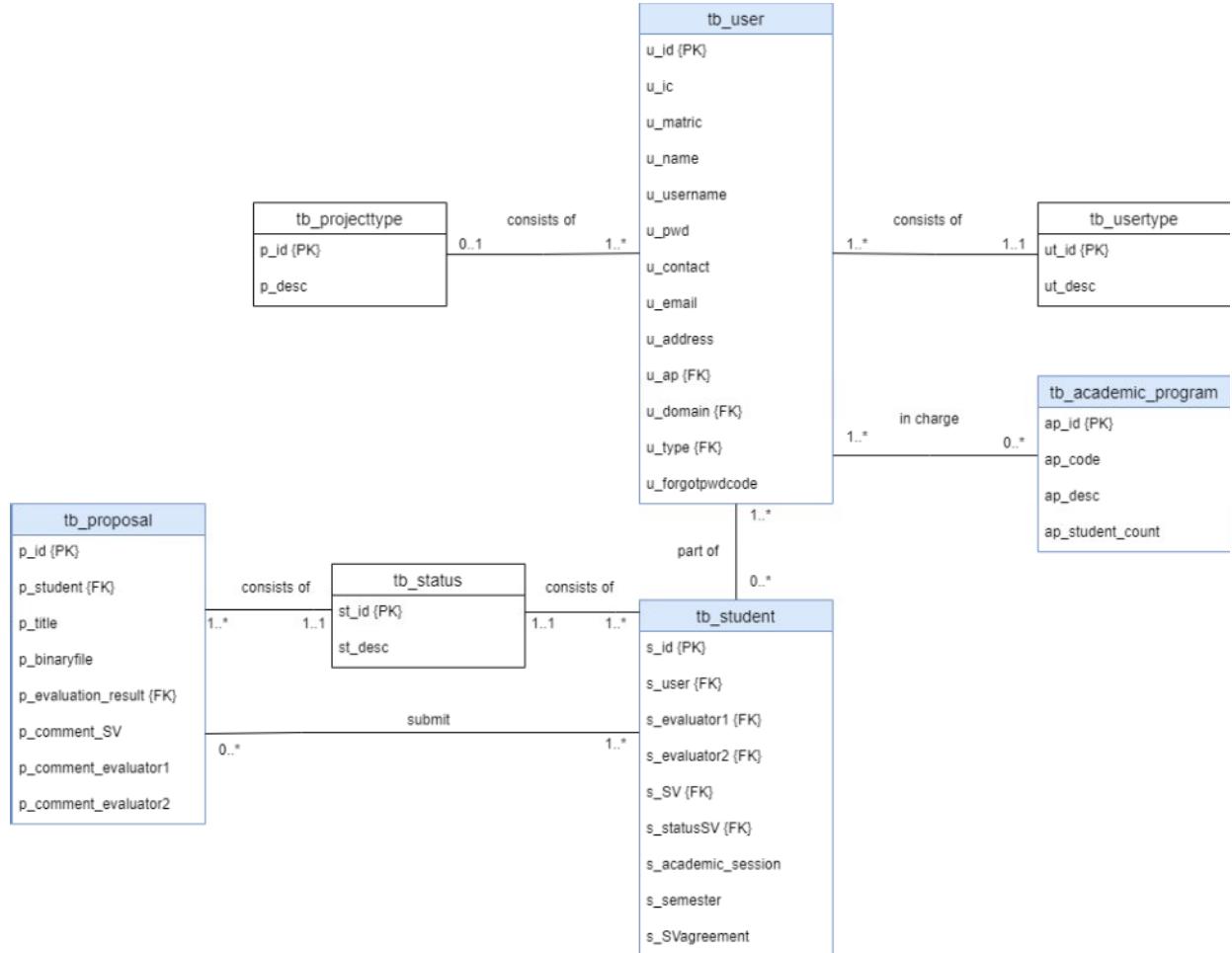


Figure 3.1 ERD for FYP1 System

ERD is the first step before the coding part since I want to have a plan for the database and think of the structure of the system and flow of content as well. There are 7 tables, blue tables representing the major modules in this AD test whereas the white tables represent the status or types for the blues tables.

Database Design

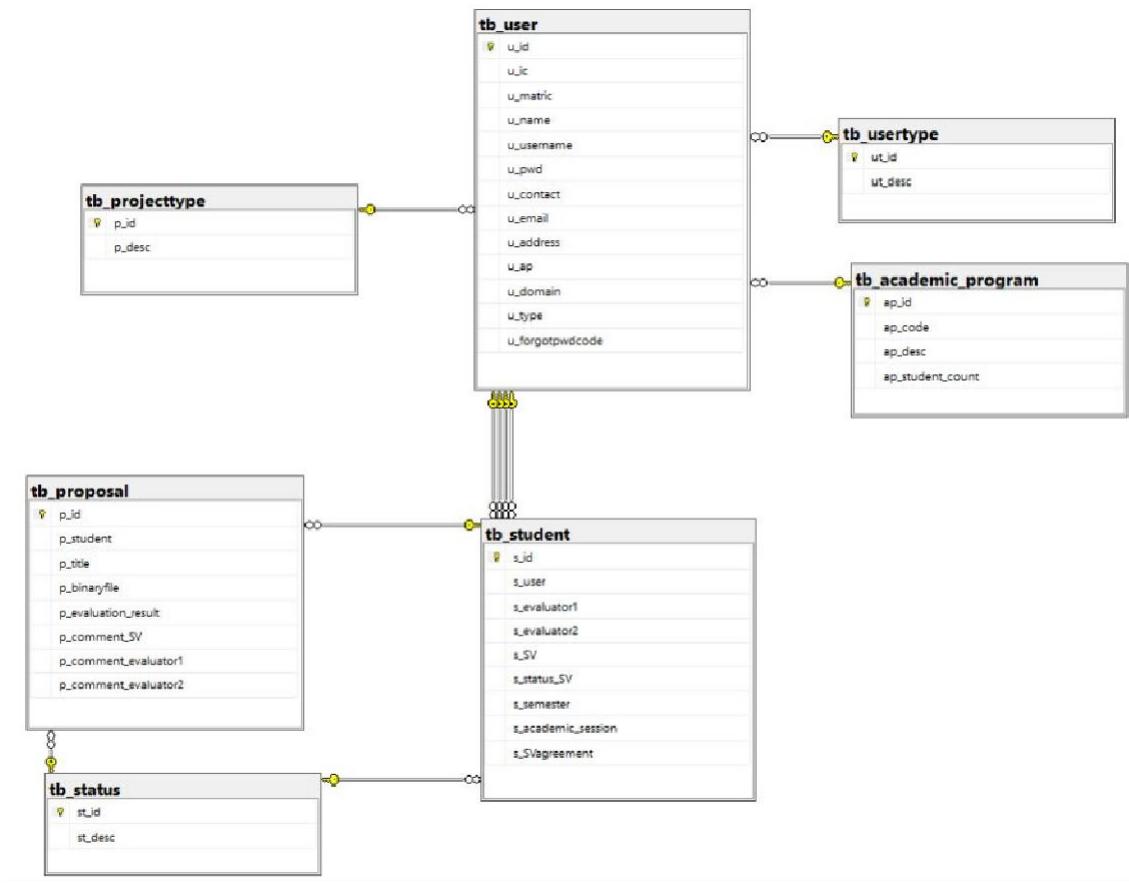
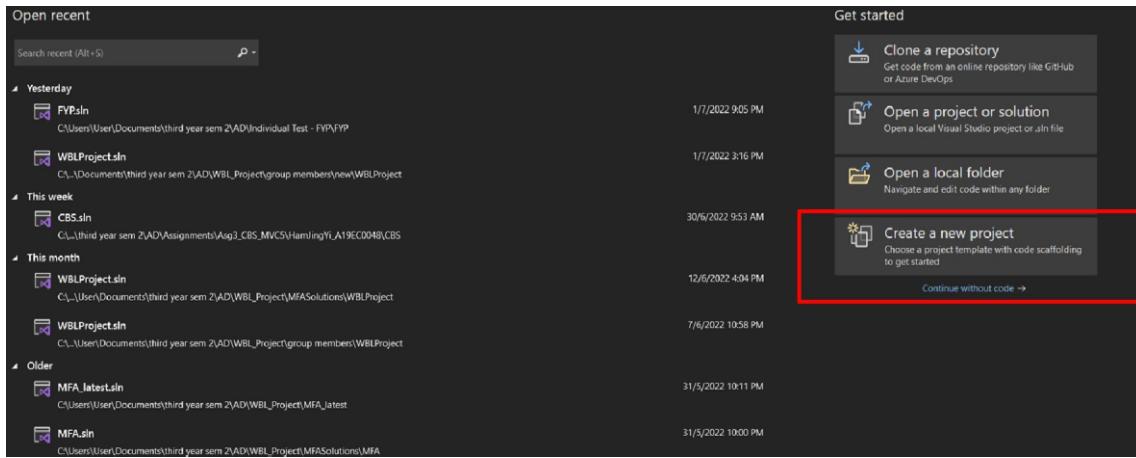


Figure 3.2 Database Design for FYP1 System

The database in SSMS is designed based on the ERD. There are four relationships between the tb_user and tb_students because the s_user, s_evaluator1, s_evaluator2, and s_SV will keep the value taken from the primary key in tb_user. The p_student attributes in tb_proposal will keep the value taken from the primary key of tb_student. p_binaryfiles in tb_proposal and s_SVagreement in tb_student will keep the filename of proposal uploaded by students and the agreements uploaded by supervisor respectively. The u_forgotpwdcode in tb_user is used in the forgot password feature on the login page.

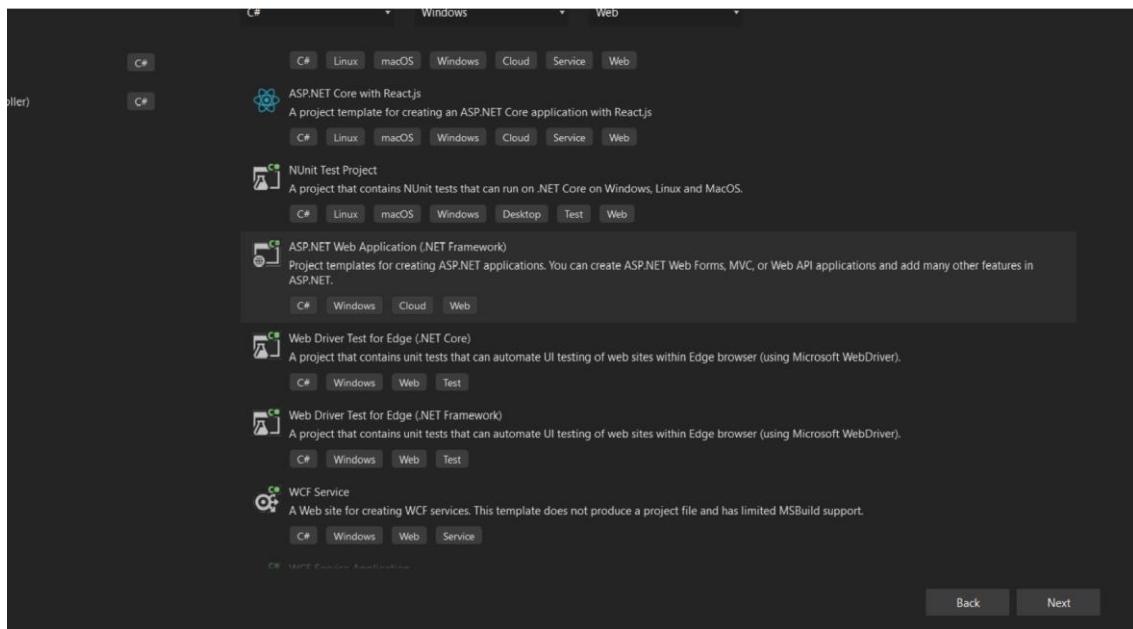
3.2 Start new project using .NET web application (C#) with MVC5

Open the Microsoft Visual Studio 2022, select and click the “Create a new project”.



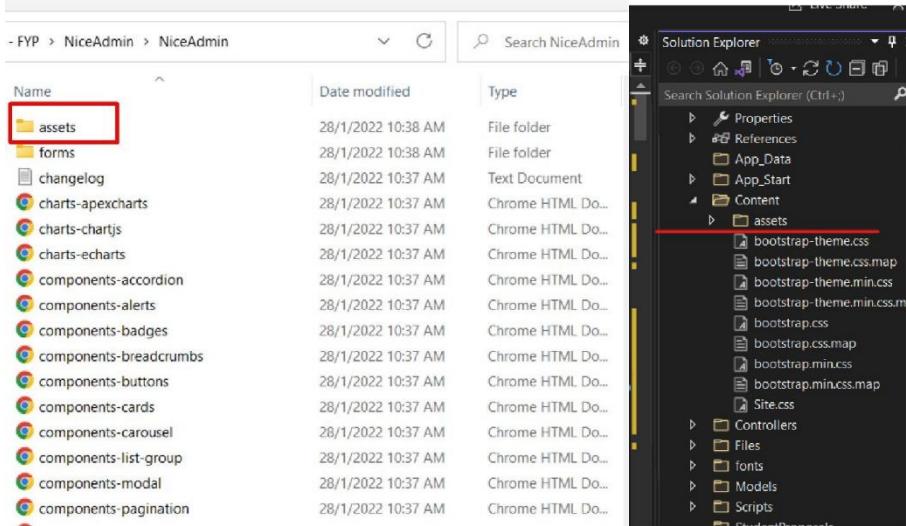
ASP.NET Web Application(.NET Framework) with MVC5 is chosen and click “Next”.

Rename the project and a new project is created.



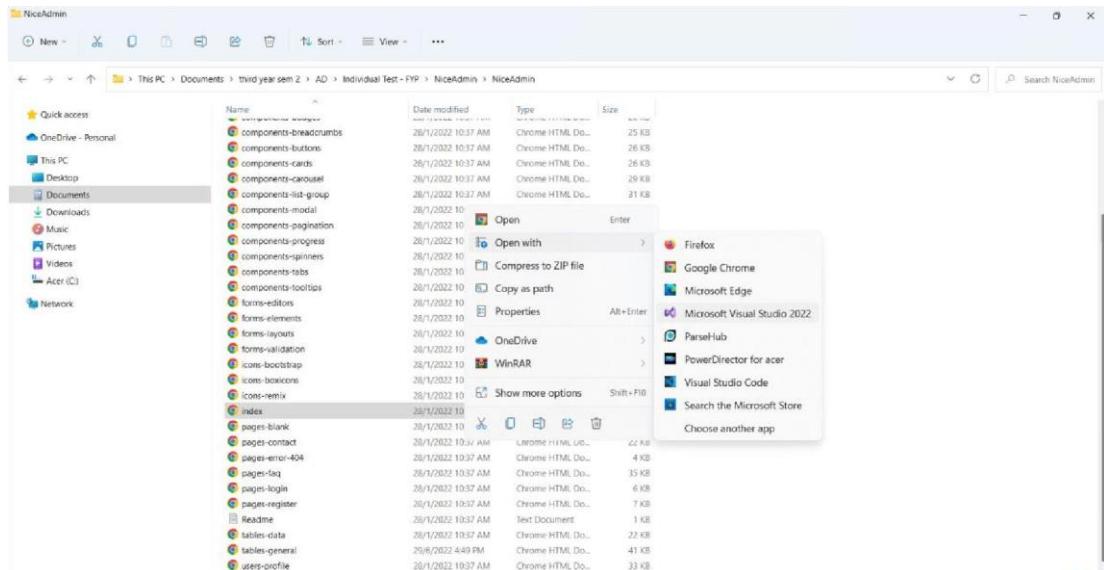
3.3 Add bootstrap template into asp.net project

After the Bootstrap template (NiceAdmin) is downloaded, unzip the file, drag the assets folder in the NiceAdmin into the Content folder of the asp.net project.



3.3.1 Change the layout and add in datatable features

Open the index.html inside the NiceAdmin using the Microsoft Visual Studio, copy the code and paste into the layout, modify to suit our FYP1 system.



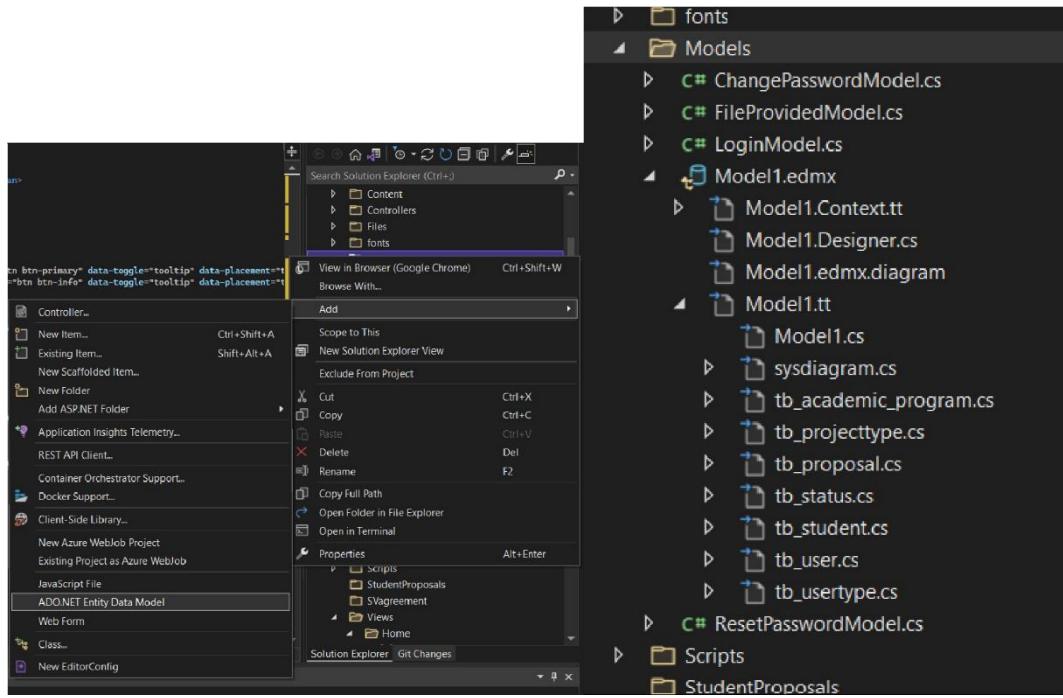
```
4 <head>
5   <title>FYPI System</title>
6   <RenderSection("head", required: false)>
7   <meta charset="utf-8">
8   <meta content="width=device-width, initial-scale=1.0" name="viewport">
9
10  <title>FYPI System</title>
11  <meta content="" name="description">
12  <meta content="" name="keywords">
13
14  <!-- Favicons -->
15  <link href="/Content/assets/img/favicon.png" rel="icon">
16  <link href="/Content/assets/img/apple-touch-icon.png" rel="apple-touch-icon">
17
18  <!-- Google Fonts -->
19  <link href="https://fonts.gstatic.com" rel="preconnect">
20  <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i|Nunito:300,300i,400,400i,600,600i" rel="stylesheet">
21
22  <!-- Vendor CSS Files -->
23  <link href="/Content/assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
24  <link href="/Content/assets/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">
25  <link href="/Content/assets/vendor/bxicons/bxicons.min.css" rel="stylesheet">
26  <link href="/Content/assets/vendor/quill/quill.snow.css" rel="stylesheet">
27  <link href="/Content/assets/vendor/quill/quill.bubble.css" rel="stylesheet">
28  <link href="/Content/assets/vendor/remixicon/remixicon.css" rel="stylesheet">
29  <link href="/Content/assets/vendor/simple-datatables/style.css" rel="stylesheet">
30
31  <script src="/Scripts/jquery-3.1.1.min.js"></script>
32
33  <!-- Template Main CSS File -->
34  <link href="/Content/assets/css/style.css" rel="stylesheet">
35
```

Modify the path inside the `<head>` tag according to where we store the asset folder.

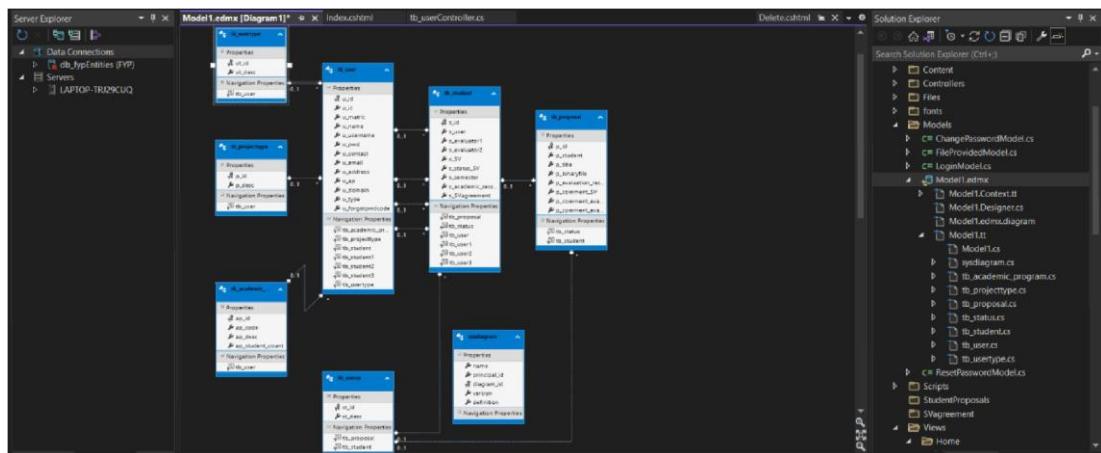
```
221      <!-- ===== Footer ===== -->
222      <footer id="footer" class="footer" style="bottom: 0; position: fixed; width: 90%; background-color :#f6f9ff;">
223          <div class="copyright">
224              &copy; Designed by <strong><span>Ham Jing Yi, Data Engineering (UTM)</span></strong>
225          </div>
226      </footer><!-- End Footer -->
227
228
229      <a href="#" class="back-to-top d-flex align-items-center justify-content-center"><i class="bi bi-arrow-up-short"></i>
230
231      <!-- Vendor JS Files -->
232      <script src="~/Content/assets/vendor/apexcharts/apexcharts.min.js"></script>
233      <script src="~/Content/assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
234      <script src="~/Content/assets/vendor/chart.js/chart.min.js"></script>
235      <script src="~/Content/assets/vendor/echarts/echarts.min.js"></script>
236      <script src="~/Content/assets/vendor/quill/quill.min.js"></script>
237      <script src="~/Content/assets/vendor/simple-datatables/simple-datatables.js"></script>
238      <script src="~/Content/assets/vendor/tinymce/tinymce.min.js"></script>
239      <script src="~/Content/assets/vendor/php-email-form/validate.js"></script>
240
241      <script src="~/Scripts/jquery-3.4.1.js"></script>
242      <script src="~/Scripts/jquery.validate.js"></script>
243      <script src="~/Scripts/jquery.validate.unobtrusive.js"></script>
244
245      <!-- Template Main JS File -->
246      <script src="~/Content/assets/js/main.js"></script>
247
248  </body>
```

Modify the path at the bottom part of the code according to where we store the asset folder.

3.4 Add in Entity Data Model and all related controllers

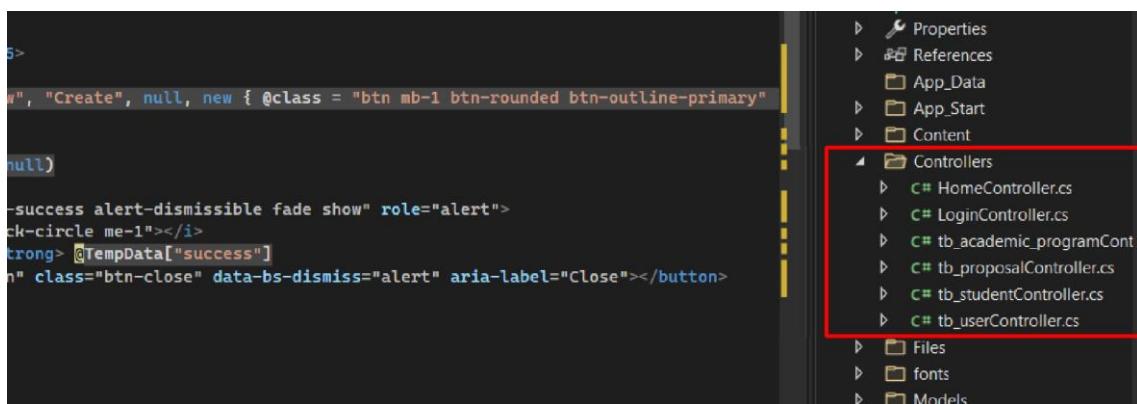


Next, we have to right click the Model and add ADO.NET Entity Data Model. The model will be added by choosing the db_fyp in SSMS. After adding successfully, all of the tables created in SSMS will occur and be stored in the Model1.tt, double click the Model1.edmx, we can view the model diagram from the database.



Then, we can add in all of the needed controllers. The added controllers will automatically create View folders and only the controller that needs the interaction between users and database like create, read, update and delete operations is added. The added controllers are

- LoginController
- tb_academic_programController
- tb_proposalController
- tb_studentController
- tb_userController



After adding all needed controllers, adding all of the card bodies and adding datatable features to all needed views.

3.5 Add Models and Views

3.5.1 Login & Logout

After adding the LoginController, adding view to all of the action result like

- Index.cshtml: login page
- ForgotPassword.cshtml: email entering for validation page after clicking forgot password
- loginlayout.cshtml: layout for all view files inside the Login Folder
- Register.cshtml: Student Registration page
- ResetPassword.cshtml: Reset password page after users clicking reset password link in email.

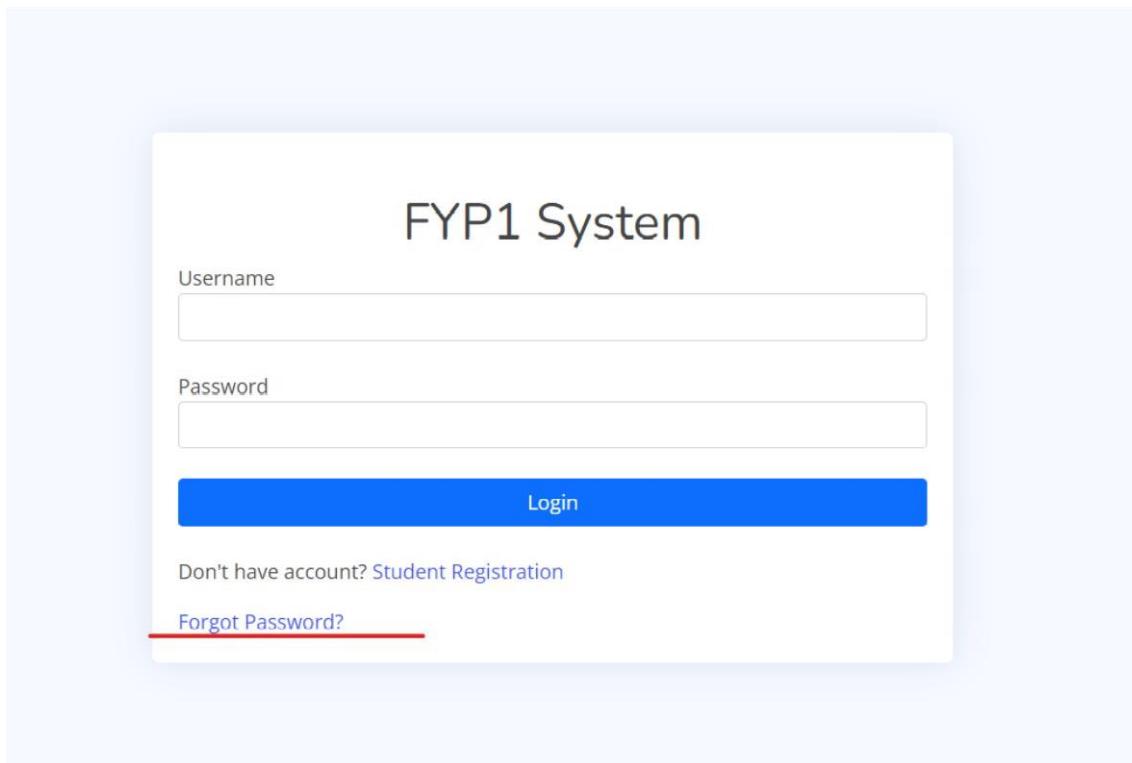
Inside the `LoginController`, there are action results for `Index` (login page) and when users enter their username and password when login to the system, this action result when check the existing user in the system, and if it is exist, store the session id, session role, session name which will be used in creating role-based views in later sections.

After users click the logout, all sessions are clear and back to the login page.

```
0 references
public ActionResult Logout()
{
    Session.Clear();
    return RedirectToAction("Index", "Login");
}
```

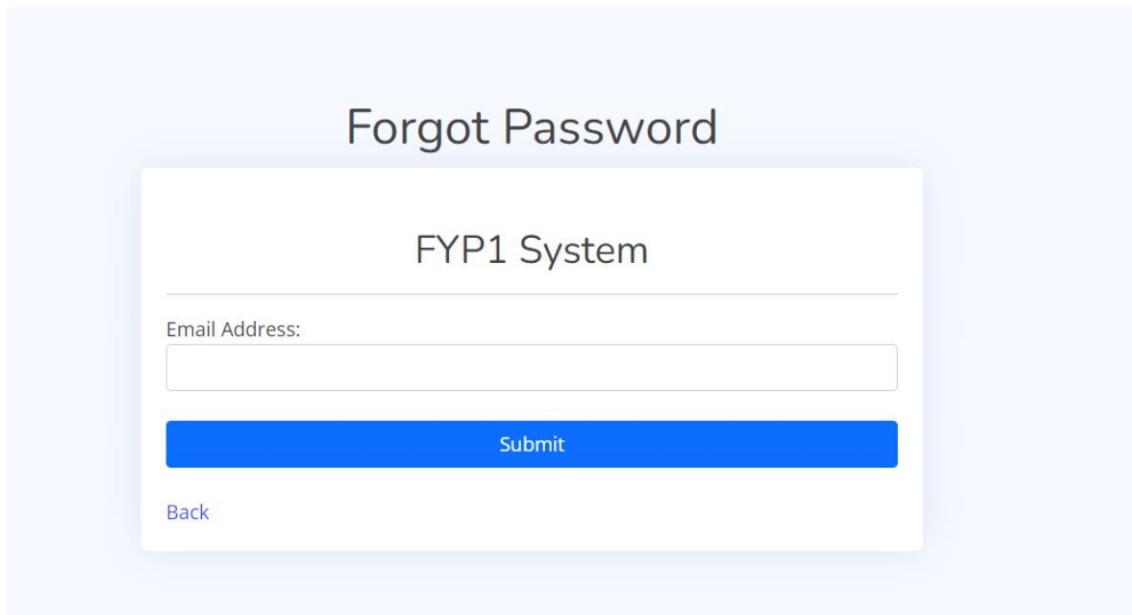
3.5.2 Forgot Password Features

Forgot Password View is added after adding the corresponding ActionResult in LoginController. After users click the forgot password link as shown below.



The screenshot shows the FYP1 System login interface. It features a light blue header with the text "FYP1 System". Below the header are two input fields: "Username" and "Password", each with a corresponding text input box. A large blue "Login" button is centered below the password field. At the bottom of the form, there are two links: "Don't have account? Student Registration" and "Forgot Password?", with the latter being underlined and having a red horizontal line underneath it.

They will be directed to the email validation page where users have to key in their email used in this system before. For lecturer/committee roles, they have to refer to the admin. Students must use the email they entered when registering.



The screenshot shows the FYP1 System forgot password interface. It has a light blue header with the text "Forgot Password". Below the header is a sub-header "FYP1 System". A text input field labeled "Email Address:" is present, with a corresponding text input box. A large blue "Submit" button is centered below the input field. At the bottom of the form, there is a "Back" link.

After the related email is found, reset password link will be sent to the entered email address and clicking into the link to reset the password. An example of reset password link is

<https://localhost:44310/Login/ResetPassword/89e200d3-ec0d-402a-bf07-517711c00b2c>

This reset password link can only use once after user manage to change password since the FYP1 System will generate unique identification number (eg: 89e200d3-ec0d-402a-bf07-517711c00b2c) for forgot password code and save into u_forgotpwdcode, this code will be append to the reset password link. After user success to change password, u_forgotpwdcode will be set null and the reset password link will invalid.

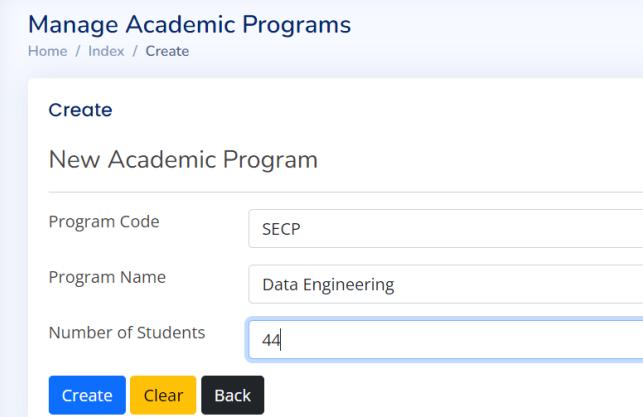
Column Name	Data Type	Allow Nulls
u_id	int	<input type="checkbox"/>
u_ic	nvarchar(12)	<input type="checkbox"/>
u_matric	nvarchar(10)	<input checked="" type="checkbox"/>
u_name	nvarchar(MAX)	<input type="checkbox"/>
u_username	nvarchar(MAX)	<input type="checkbox"/>
u_pwd	nvarchar(MAX)	<input type="checkbox"/>
u_contact	nvarchar(12)	<input checked="" type="checkbox"/>
u_email	nvarchar(MAX)	<input type="checkbox"/>
u_address	nvarchar(MAX)	<input checked="" type="checkbox"/>
u_ap	int	<input checked="" type="checkbox"/>
u_domain	int	<input checked="" type="checkbox"/>
u_type	int	<input checked="" type="checkbox"/>
u_forgotpwdcode	nvarchar(MAX)	<input checked="" type="checkbox"/>

3.6 Academic Program Module

Role access:

- **Admin** CRUD Academic Programs

Admin can manage the academic program. This module is needed to determine the academic program of a user and assign committees from lecturers based on academic programs which include **Software Engineering and Data Engineering**. When creating, admin must enter the program code, program name and number of students for the programs. Admin can edit, view details, and delete the programs.



Manage Academic Programs

Home / Index / Create

Create

New Academic Program

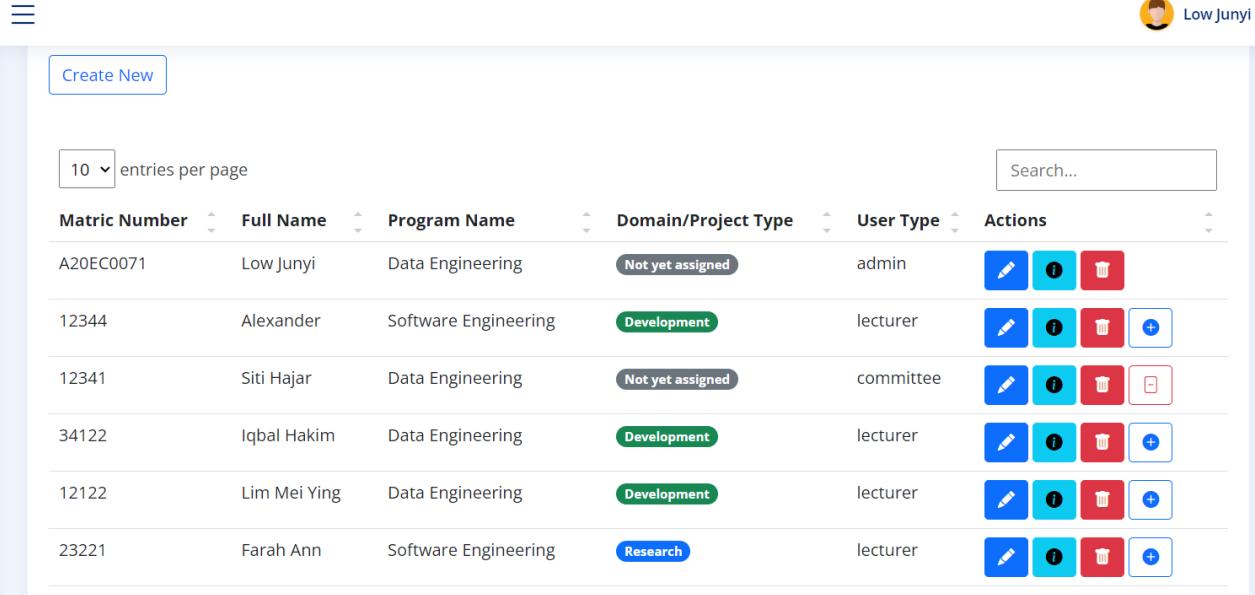
Program Code: SECP

Program Name: Data Engineering

Number of Students: 44

Create Clear Back

Admin can manage the lecturer list for each program.



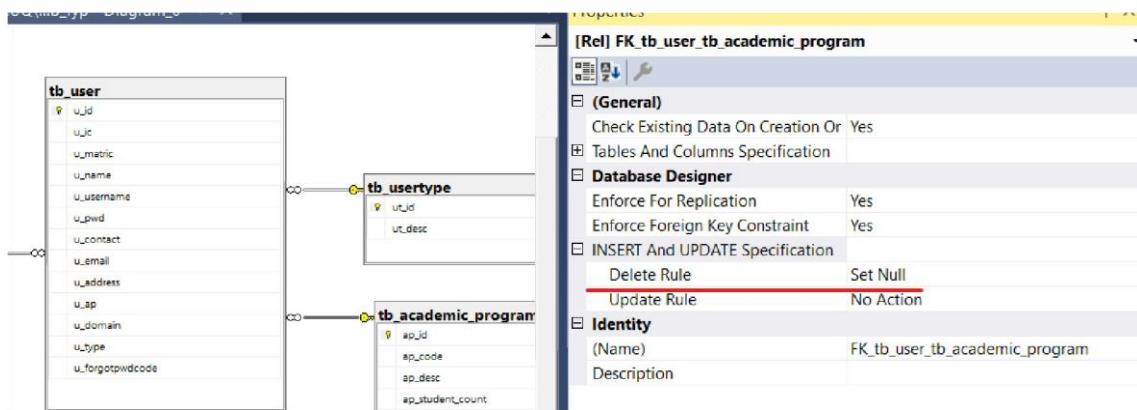
Create New

10 entries per page

Search...

Matric Number	Full Name	Program Name	Domain/Project Type	User Type	Actions
A20EC0071	Low Junyi	Data Engineering	Not yet assigned	admin	
12344	Alexander	Software Engineering	Development	lecturer	
12341	Siti Hajar	Data Engineering	Not yet assigned	committee	
34122	Iqbal Hakim	Data Engineering	Development	lecturer	
12122	Lim Mei Ying	Data Engineering	Development	lecturer	
23221	Farah Ann	Software Engineering	Research	lecturer	

After clicking the link “Lecturer List for this Academic Program”, only lecturers inside this “Data Engineering” will be shown.



The relationship between tb_user and tb_academic_program is set null because when one of the data is deleted in tb_academic_program, the column in tb_user related with the primary key in tb_academic_program will set null.

3.6.1 Create Role-Based Views

```

187 <li class="nav-item">
188   <a class="nav-link collapsed" href="/tb_proposal/">
189     <i class="bi bi-book"></i>
190     <span>All Proposals</span>
191   </a>
192 </li>
193
194
195   <div class="collapse" id="collapseAcademicPrograms">
196     <ul class="list-group list-group-flush">
197       <li class="list-group-item">
198         <a class="nav-link collapsed" href="/tb_academic_program/">
199           <i class="bi bi-book"></i>
200           <span>Academic Programs</span>
201         </a>
202       </li>
203     </ul>
204   </div>
205
206   <li class="nav-item">
207     <a class="nav-link collapsed" href="/tb_user/">
208       <i class="bi bi-diagram-3"></i>
209       <span>All Lecturers</span>
210     </a>
211   </li>
212
213 </div>

```

Only admin (Session[“role”] = 1) will be able to view the Academic Program Module at the sidebar.

3.7 Users Module

Role access:

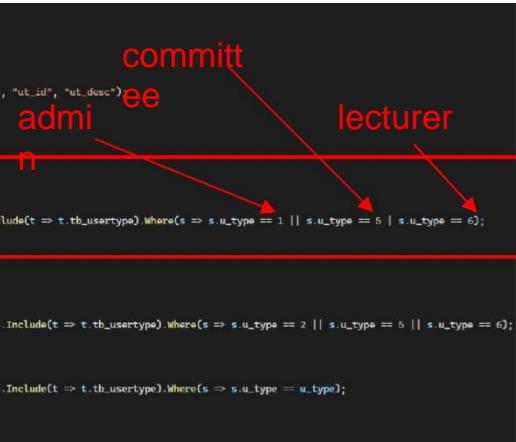
- **Admin** Manage Lecturer
 - Assign and delete committee
- **Student** registration
- **Committee** Manage Student and edit lecturer list (assign domain)

This user module will store all the common user information like IC number, matric number, full name, username, password, contact number, email address, home address, academic program, domain/project type, and user type. The action result index will be filtered by usertype in such a way that admin can only view lecturer or committee list whereas committee can view lecturer and students list. Students will register personal details using the tb_user of this module.

3.7.1 Create Role-Based Views

Admin

Index:



```
//get user session
var session = Session["role"];
if (session == null)
{
    Response.Redirect("~/Login");
}
var filtertype = new SelectList(db.tb_usertype.Where(g => g.ut_id == 2 || g.ut_id == 5 || g.ut_id == 6), "ut_id", "ut_desc");
ViewData["type"] = filtertype;
if(apid == -1)
{
    ViewData["Selected"] = -1;
}
if (Session["role"].ToString() == "1")
{
    var tb_user = db.tb_user.Include(t => t.tb_academic_program).Include(t => t.tb_projecttype).Include(t => t.tb_usertype).Where(s => s.u_type == 1 || s.u_type == 5 || s.u_type == 6);
    return View(tb_user.ToList());
}
if (Session["role"].ToString() == "5")
{
    if (u_type == 0)
    {
        var tb_user = db.tb_user.Include(t => t.tb_academic_program).Include(t => t.tb_projecttype).Include(t => t.tb_usertype).Where(s => s.u_type == 2 || s.u_type == 5 || s.u_type == 6);
        return View(tb_user.ToList());
    }
    else
    {
        var tb_user = db.tb_user.Include(t => t.tb_academic_program).Include(t => t.tb_projecttype).Include(t => t.tb_usertype).Where(s => s.u_type == u_type);
        return View(tb_user.ToList());
    }
}
else
{
    var tb_user = db.tb_user.Include(t => t.tb_academic_program).Include(t => t.tb_projecttype).Include(t => t.tb_usertype).Where(s => s.u_type == 1 || s.u_type == 5 || s.u_type == 6);
    return View(tb_user.ToList());
}
```

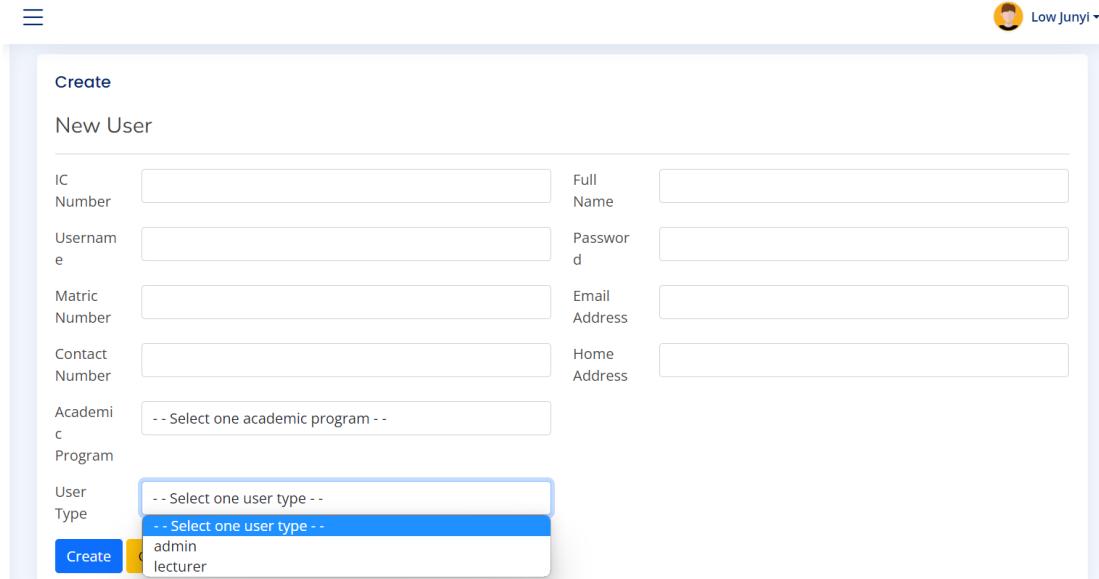
At the ActionResult Index inside the tb_userController, admin is allowed to view the lecturer and committee list. This is because the admin has to assign committees from the lecturer list.

Create:

```
public ActionResult Create()
{
    ViewBag.u_ap = new SelectList(db.tb_academic_program, "ap_id", "ap_desc");
    ViewBag.u_domain = new SelectList(db.tb_projecttype, "p_id", "p_desc");
    if(Session["role"].ToString() == "1")
    {
        ViewBag.u_type = new SelectList(db.tb_usertype.Where(t => t.ut_id == 1 || t.ut_id == 6), "ut_id", "ut_desc");
    }
}
```

admin lecturer

The SelectList of usertype will display the role of admin and lecturer so that admin can create a user of role admin or lecturer.



Three red arrows point from the text "lecturer" in the previous code block to the "lecturer" option in the dropdown menu on the screenshot.

Three red arrows point from the text "admin" in the previous code block to the "admin" option in the dropdown menu on the screenshot.

Three red arrows point from the text "admin" and "lecturer" in the previous code block to the "admin" and "lecturer" labels above the dropdown menu on the screenshot.

Figure above is the create form in the user module from the view of the admin. There is no domain field since admin will assign a committee and that committee will assign the domain to the lecturers created by admin.

Edit:

```
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    tb_user tb_user = db.tb_user.Find(id);
    if (tb_user == null)
    {
        return HttpNotFound();
    }

    ViewBag.u_ap = new SelectList(db.tb_academic_program, "ap_id", "ap_desc", tb_user.u_ap);
    ViewBag.u_domain = new SelectList(db.tb_projecttype, "p_id", "p_desc", tb_user.u_domain);

    if (Session["role"].ToString() == "1")
    {
        ViewBag.u_type = new SelectList(db.tb_usertype.Where(t => t.ut_id == 1 || t.ut_id == 5 || t.ut_id == 6), "ut_id", "ut_desc", tb_user.u_type);
    }
    if (Session["role"].ToString() == "5")
    {
        ViewBag.u_type = new SelectList(db.tb_usertype.Where(t => t.ut_id == 2), "ut_id", "ut_desc", tb_user.u_type);
    }
    return View(tb_user);
}
```

committee

The SelectList of usertype will display the role of admin, committee and lecturer so that admin can assign a committee based on the lecturer and academic programs.

Manage Lecturers

Home / Index / Edit

Edit

User

IC Number: 769090651234

Username: limmei

Matric Number: 12122

Contact Number: 0124663976

Academic Program: Data Engineering

User Type: lecturer

Save Cancel

-- Select one user type --
admin
committee
lecturer

Below is the alternative way for the admin to assign/remove the committee.

Assign committee role:

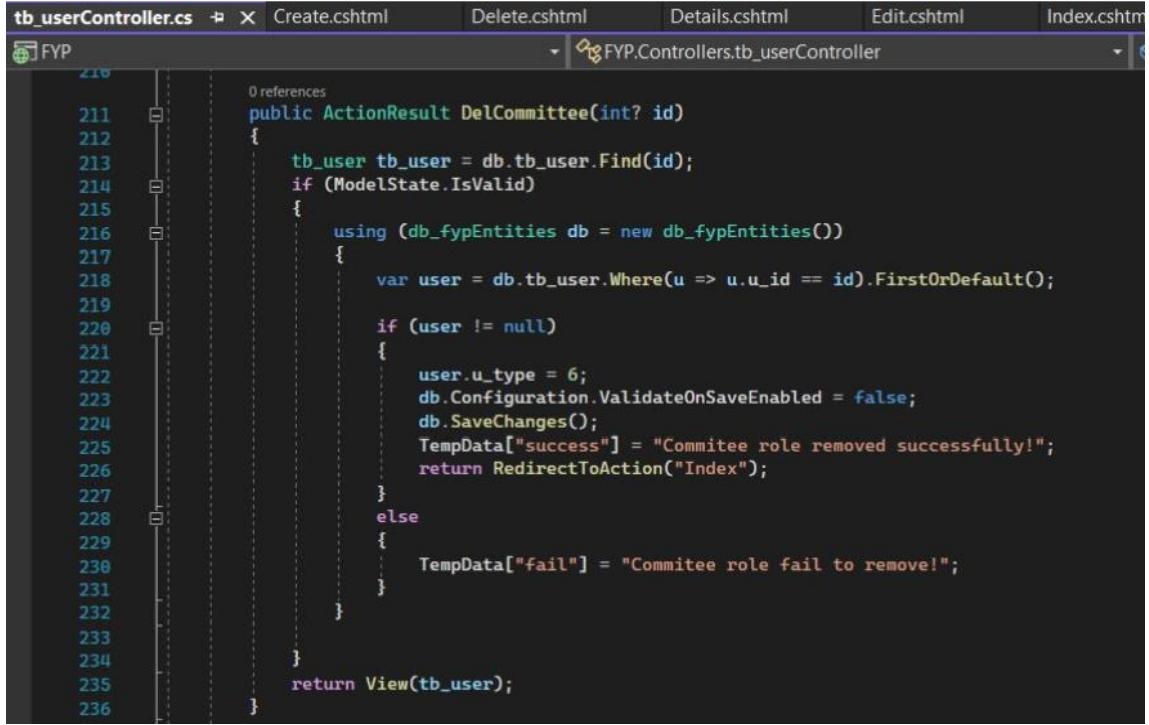
```

tb_userController.cs  X  Create.cshtml  Delete.cshtml  Details.cshtml  Edit.cshtml  Index.cshtml
FYP  FYP.Controllers.tb_userController

184  public ActionResult AddCommittee(int? id)
185  {
186      tb_user tb_user = db.tb_user.Find(id);
187      if (ModelState.IsValid)
188      {
189          using (db_fypEntities db = new db_fypEntities())
190          {
191              var user = db.tb_user.Where(u => u.u_id == id).FirstOrDefault();
192
193              if (user != null)
194              {
195                  user.u_type = 5;
196                  db.Configuration.ValidateOnSaveEnabled = false;
197                  db.SaveChanges();
198                  TempData["success"] = "Committee assigned successfully!";
199                  return RedirectToAction("Index");
200              }
201              else
202              {
203                  TempData["fail"] = "Committee fail to assign!";
204              }
205          }
206
207      }
208
209      return View(tb_user);
210  }

```

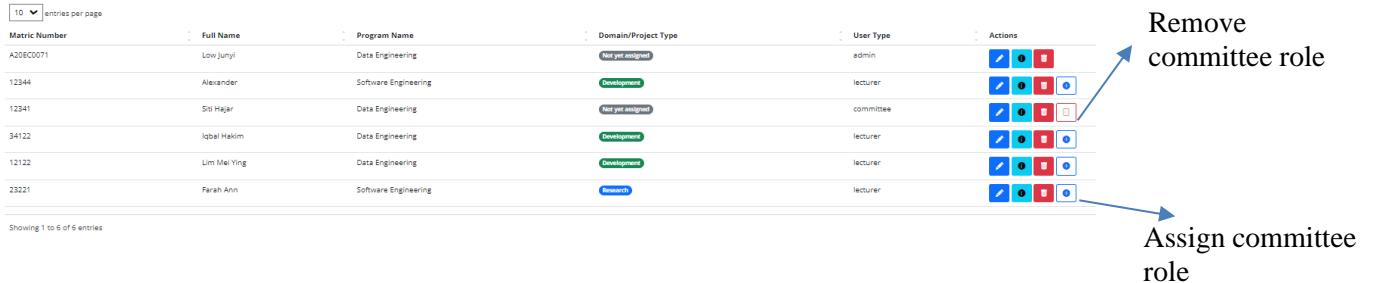
Remove committee role:



```
tb_userController.cs  Create.cshtml  Delete.cshtml  Details.cshtml  Edit.cshtml  Index.cshtml
FYP  FYP.Controllers.tb_userController

211  public ActionResult DelCommittee(int? id)
212  {
213      tb_user tb_user = db.tb_user.Find(id);
214      if (ModelState.IsValid)
215      {
216          using (db_fypEntities db = new db_fypEntities())
217          {
218              var user = db.tb_user.Where(u => u.u_id == id).FirstOrDefault();
219
220              if (user != null)
221              {
222                  user.u_type = 6;
223                  db.Configuration.ValidateOnSaveEnabled = false;
224                  db.SaveChanges();
225                  TempData["success"] = "Committee role removed successfully!";
226                  return RedirectToAction("Index");
227              }
228              else
229              {
230                  TempData["fail"] = "Committee role fail to remove!";
231              }
232          }
233      }
234
235      return View(tb_user);
236  }
```

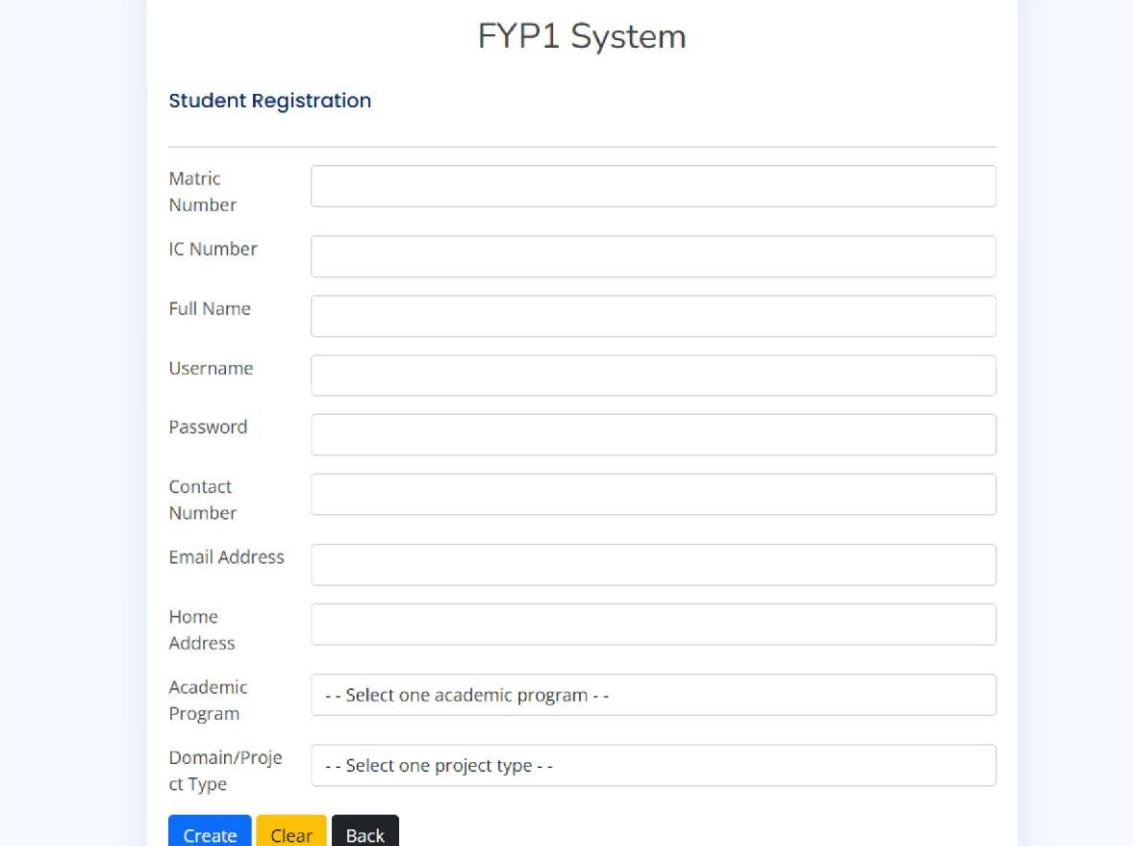
I have created two new ActionResult of AddCommittee and DelCommittee in such a way that admin can directly assign a lecturer as committee or remove committee role and back to lecturer role at the Index page without clicking the edit button.



Matrix Number	Full Name	Program Name	Domain/Project Type	User Type	Actions
A208C0071	Lov Junyl	Data Engineering	Not yet assigned	admin	  
12344	Alexander	Software Engineering	Development	lecturer	  
12341	Siti Hajar	Data Engineering	Not yet assigned	committee	  
34122	Iqbal Hakim	Data Engineering	Development	lecturer	  
12122	Lim Mei Ying	Data Engineering	Development	lecturer	  
23221	Farah Ann	Software Engineering	Research	lecturer	  

Student

When students register their personal information by clicking the “Student Registration” at the login page, students are unable to choose the usertype, instead the usertype will be set as hidden field and hidden post value is 2 (student role).



FYP1 System

Student Registration

Matric Number

IC Number

Full Name

Username

Password

Contact Number

Email Address

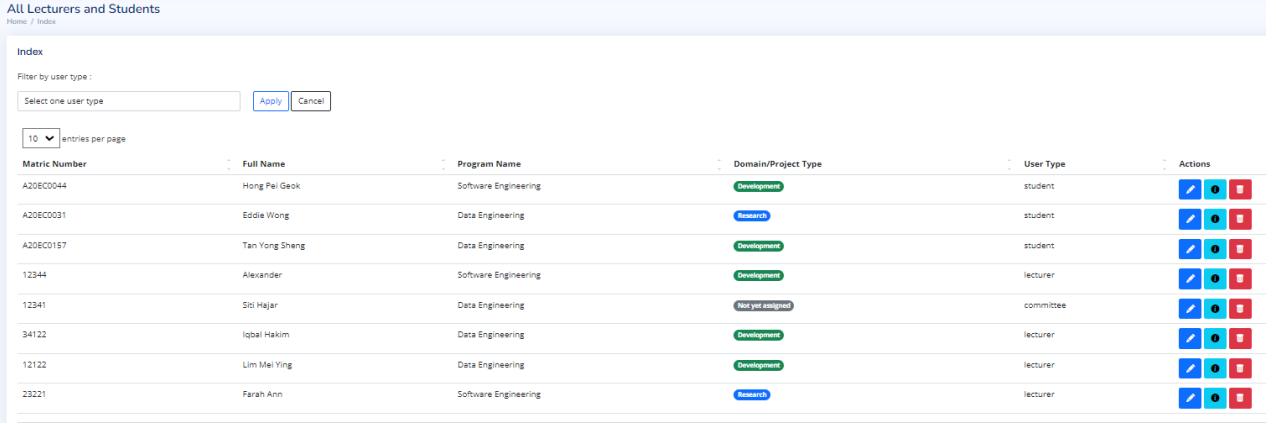
Home Address

Academic Program

Domain/Project Type

Create **Clear** **Back**

The information will be saved into tb_user and the committee will be able to view the complete student list in the index page of “All lecturers and students” menu on the sidebar.



All Lecturers and Students							
Index							
Filter by user type :							
Select one user type		Apply	Cancel				
10 entries per page							
Matric Number	Full Name	Program Name	Domain/Project Type	User Type	Actions		
A20EC0044	Hong Pei Geok	Software Engineering	Development	student			
A20EC0031	Eddie Wong	Data Engineering	Research	student			
A20EC0157	Tan Yong Sheng	Data Engineering	Development	student			
12344	Alexander	Software Engineering	Development	lecturer			
12341	Siti Hajar	Data Engineering	Not yet assigned	committee			
34122	Iqbal Hakim	Data Engineering	Development	lecturer			
12122	Lim Mei Ying	Data Engineering	Development	lecturer			
23221	Farah Ann	Software Engineering	Research	lecturer			

Committee

Index:

```
var filtertype = new SelectList(db.tb_usertype.Where(g => g.ut_id == 2 || g.ut_id == 5 || g.ut_id == 6), "ut_id", "ut_desc");
ViewData["type"] = filtertype;
if(apid == -1)
{
    ViewData["back"] = null;
    if (Session["role"].ToString() == "1")
    {
        var tb_user = db.tb_user.Include(t => t.tb_academic_program).Include(t => t.tb_projecttype).Include(t => t.tb_usertype).Where(s => s.u_type == 1 || s.u_type == 2 || s.u_type == 6);
        return View(tb_user.ToList());
    }
    if (Session["role"].ToString() == "5")
    {
        if (u_type == 0)
        {
            var tb_user = db.tb_user.Include(t => t.tb_academic_program).Include(t => t.tb_projecttype).Include(t => t.tb_usertype).Where(s => s.u_type == 2 || s.u_type == 5 || s.u_type == 6);
        }
        else
        {
            var tb_user = db.tb_user.Include(t => t.tb_academic_program).Include(t => t.tb_projecttype).Include(t => t.tb_usertype).Where(s => s.u_type == u_type);
        }
    }
}
```

committ
ee
student
lecturer
filter

At the ActionResult Index inside the tb_userController, the committee is allowed to view the lecturer, committee and student list. The committee list is also available since the committee is also the lecturer and domain can be assigned.

Edit (Assign Domain):

All Lecturers and Students

Home / Index / Edit

Edit

User

IC Number	010308141213	Full Name	Hong Pei Geok
Username	hongpeigeok	Email Address	low.junyi2001@gmail.com
Matric Number	A20EC0044	Home Address	KDSE, UTM
Contact Number	0182858291	Domain/Project Type	development
Academic Program	Software Engineering		

Save Cancel

Since the committee have to manage students and lecturers, I have provide a filter for committee to filter the user list by user type.

Index

Filter by user type :

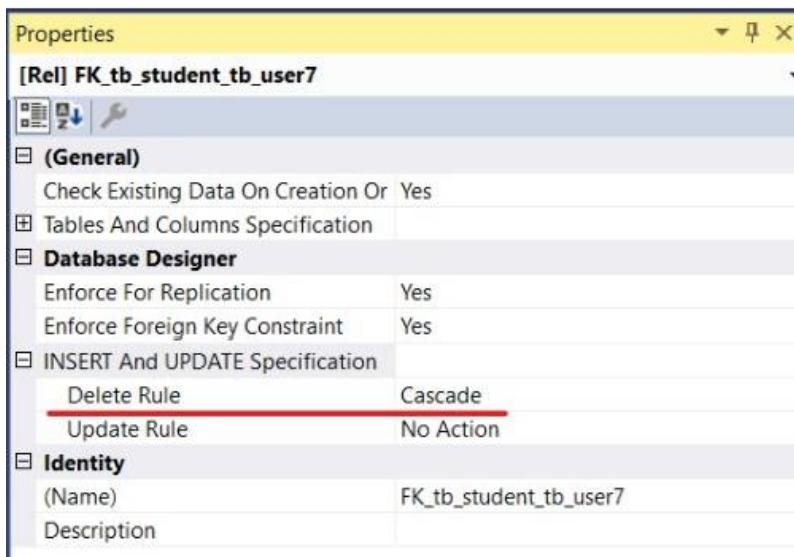
Select one user type

Apply

Cancel

Matric Number	Full Name	Program Name
A20EC0044	Hong Pei Geok	Software Engineering
A20EC0031	Eddie Wong	Data Engineering
A20EC0157	Tan Yong Sheng	Data Engineering
12344	Alexander	Software Engineering
12341	Siti Hajar	Data Engineering
34122	Iqbal Hakim	Data Engineering
12122	Lim Mei Ying	Data Engineering
23221	Farah Ann	Software Engineering

Delete:



Delete rule of relationship between tb_user and tb_student is set to cascade, meaning that if the user role of lecturer or committee in the tb_user is deleted, the **related** row of data will also be deleted in the tb_student and students need to create (select new supervisor) for their proposals.

3.8 PIC in Proposal Module

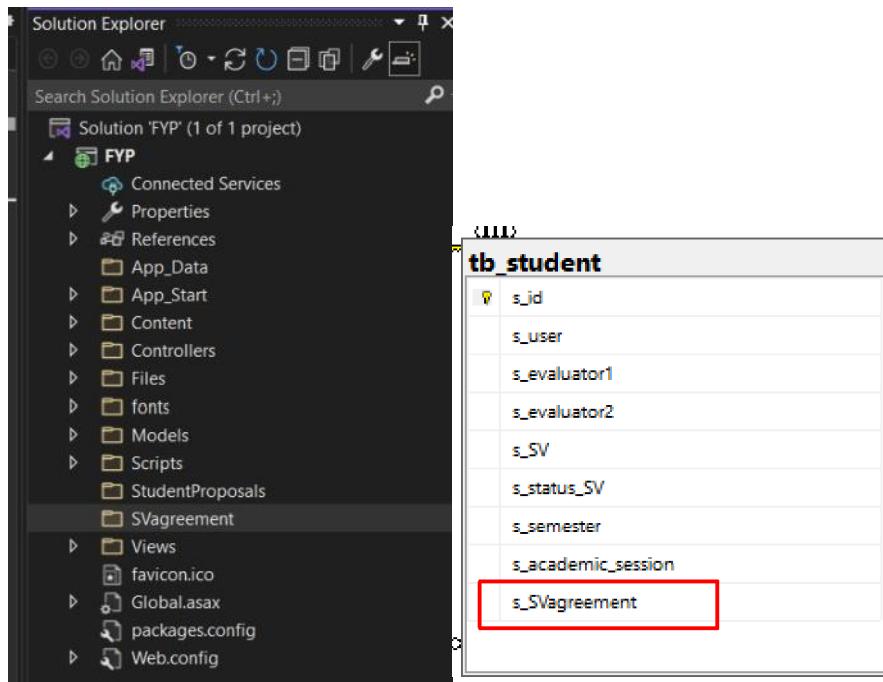
Role access:

- **Student** select supervisor
- **Committee** view the selected supervisor and assign two evaluators
- **Lecturer (Supervisor)** view own students, upload agreement

This PIC in Proposal Module will store the information of all persons in charge of the students proposal such as student id, evaluator 1, evaluator 2, supervisor, status of preferred supervisor, semester, academic session and agreement filename. The action result index will be filtered by academic session and semester of students in such a way that lecturer (Supervisor) can only view own students based on academic session and semester whereas the committee can see all students who apply for supervisor. Before that, students have to select a supervisor and fill in all needed information in the creation process.

3.8.1 Add a folder for Supervisor Agreement

In this module, supervisors have to upload an agreement if they agree students to select them as supervisor. Therefore, a new folder name named “SVagreement” is created to store all of the uploaded agreement PDF file whereas the s_SVagreement in the tb_student is used to store the filename of the uploaded PDF file.



The supervisor can download the agreement form in doc or pdf form on the home page, fill in all needed information and upload back to the system.

The screenshot shows the FYP1 System home page. On the left, there is a sidebar with links: Home (highlighted with a red box), Students, and All Proposals. The main content area is titled 'FYP1 Forms' and contains a list of download links:

- [Project-Proposal-Form.doc](#)
- [Project-Proposal-Form.pdf](#)
- [Supervision-General-Agreement.doc](#) (highlighted with a red box)
- [Supervision-General-Agreement.pdf](#)

3.8.2 Create Role-Based Views

Student

Create (select supervisor):

Students are the ones who create and select their preferred supervisor. The items in the drop down list for selecting the supervisor is set in such a way that students can only select supervisors who have the same domain/project type with students and students can only choose users with lecturer or committee role to be their supervisor.

```

155 // GET: tb_student/Create
156 public ActionResult Create()
157 {
158     var session = Session["role"];
159     if (session == null)
160     {
161         Response.Redirect("~/Login");
162     }
163     var num = Session["domain"].ToString();
164     int domain = Convert.ToInt32(num);
165
166     ViewBag.s_pid = new SelectList(db.tb_proposal, "p_id", "p_title");
167     ViewBag.s_status_SV = new SelectList(db.tb_status, "st_id", "st_desc");
168     ViewBag.s_SV = new SelectList(db.tb_user.Where(g => g.u_type != 1 && g.u_type != 2 && g.u_domain == domain).ToList(), "u_id", "u_name");
169     ViewBag.s_user = new SelectList(db.tb_user, "u_id", "u_name", Session["id"]);
170     ViewBag.s_evaluator1 = new SelectList(db.tb_user.Where(g => g.u_type == 6).ToList(), "u_id", "u_name");
171     ViewBag.s_evaluator2 = new SelectList(db.tb_user.Where(g => g.u_type == 6).ToList(), "u_id", "u_name");
172
173     return View();
174 }

```

If a student does not choose a supervisor before, “select supervisor” button will appear.

Index						
Select Supervisor						
10	▼	entries per page	<input type="text" value="Search..."/>			
Academic Session	Semester	Preferred Supervisor	Agreement	Evaluator 1	Evaluator 2	Actions

Else, “Select Supervisor” button will not appear. If the status of preferred supervisor is rejected or pending, students can delete the record and select a new supervisor.

Academic Session	Semester	Preferred Supervisor	Agreement	Evaluator 1	Evaluator 2	Actions	
2022/2023	1	Lim Mei Ying	Approved	limmei_yunzheng.pdf	Alexander	Iqbal Hakim	

Index:

Students can only own data. This is done by taking the session id of the student and using the where clause to filter.



```
index.cshtml tb_studentController.cs X
FYP FYP.Controllers.tb_studentController filename
92     }
93     else
94     {
95         if (Session["role"].ToString() == "2")
96         {
97             var stu = Session["id"].ToString();
98             int p = Convert.ToInt32(stu);
99             var tb_student = db.tb_student.Include(t => t.tb_proposal).Where(s => s.s_user == p).Include(t => t.tb_status).Include(t => t.tb_user).Incl
100            if (tb_student.Count() > 0)
101            {
102                ViewData["create"] = null;
103            }
104            else
105            {
106                ViewData["create"] = 1;
107            }
108            return View(tb_student.ToList());
109        }

```

Students can only view and download the agreement once SV upload agreement and approved by committee. If the status is pending or rejected, the student is able to delete the selected supervisor. Once the supervisor upload agreement and approved by committee, students no longer can delete the data.

Academic Session	Semester	Preferred Supervisor	Agreement	Evaluator 1	Evaluator 2	Actions
2022/2023	2	Farah Ann	Pending	Not Available		 

If the status is Pending, students can only view agreement after the committee approved the preferred supervisor.

Delete:

Students can only delete the selected supervisor if the status of the selected supervisor is pending or rejected.

PIC in Proposal
Home / Index / Delete

Delete

Are you sure you want to delete this?

Academic Session (eg. 2022/2023)	2022/2023
Semester (eg. 1)	2
Student	Eddie Wong
Supervisor	Farah Ann
Supervisor Agreement	Not Available
Supervisor Approval Status	Pending
Evaluator 1	
Evaluator 2	

Delete **Back**

Index

 **Success!** Information deleted successfully! X

Select Supervisor

10 entries per page

Academic Session	Semester	Preferred Supervisor	Agreement	Evaluator 1	Evaluator 2	Actions
No entries found						

Lecturer (Supervisor/Evaluator)

View own students:

If the lecturers are selected as one of the PIC for students after students select the supervisor, the PIC can only view their own students once selected.

```
else
{
    var stu = Session["id"].ToString();
    int p = Convert.ToInt32(stu);

    var tb_student = db.tb_student
        .Include(t => t.tb_proposal)
        .Include(t => t.tb_status)
        .Include(t => t.tb_user)
        .Include(t => t.tb_user1)
        .Include(t => t.tb_user2)
        .Include(t => t.tb_user3)
        .Where(s => (s.s_SV == p || s.s_evaluator1 == p || s.s_evaluator2 == p) && (s.s_academic_session == s_academic_session && s.s_semester == s_semester))

    return View(tb_student.ToList());
}
```

View own
students

Filters

Filter is provided so that the evaluator / supervisor can view own students based on semester and academic session.

[Index](#)

Filter by Academic Session and Semester

Select Academic Sessions	Select Semesters	Apply	Cancel				
10 entries per page	Search...						
Academic Session	Semester	Student	Preferred Supervisor	Agreement	Evaluator 1	Evaluator 2	Actions
2022/2023	1	Hong Pei Geok	Lim Mei Ying Approved	limmei_yunzheng.pdf	Alexander	Iqbal Hakim	  
2022/2023	1	Tan Yong Sheng	Alexander Approved	alex_sofia.pdf	Iqbal Hakim	Lim Mei Ying	  

Edit (Supervisor upload agreement):

The supervisor can view the students once students select lecturers to be their supervisor. If the agreement column is blank, the supervisor can upload their agreement by clicking the yellow upload agreement button if they agree to be supervisor of the students.

Academic Session	Semester	Student	Preferred Supervisor	Agreement	Evaluator 1	Evaluator 2	Actions
2022/2023	1	Hong Pei Geok	Lim Mei Ying	limmei_yunzheng.pdf	Alexander	Iqbal Hakim	  
2022/2023	1	Tan Yong Sheng	Alexander	alex_sofia.pdf	Iqbal Hakim	Lim Mei Ying	  
2022/2023	1	Eddie Wong	Farah Ann	Pending			  

Supervisors can upload their agreement in PDF form.

Agreement

Agreement

Supervisor Agreement No file chosen

Save **Back**

If the supervisor does not submit anything and click the Save button, a warning will be displayed and no data is saved.

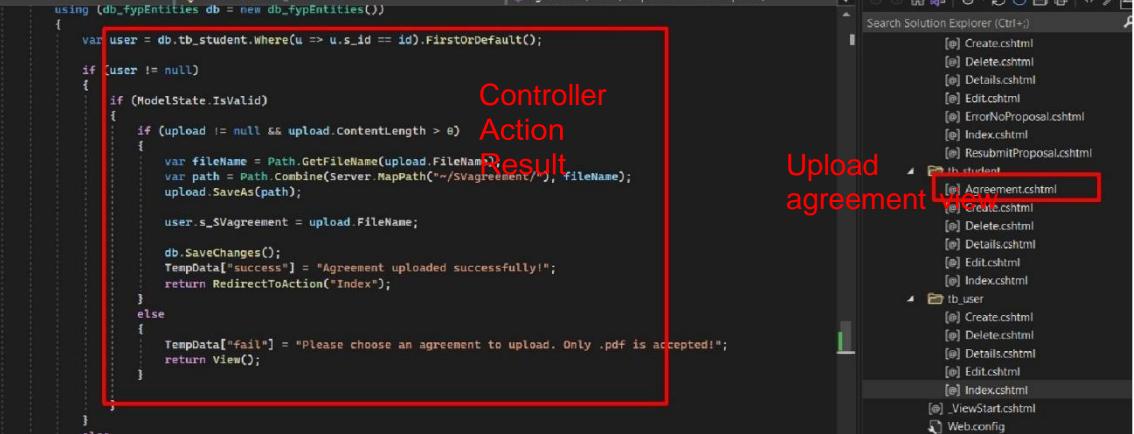
Agreement

⚠ Warning! Please choose an agreement to upload. Only .pdf is accepted! X

Agreement

Supervisor Agreement No file chosen

Save **Back**



```

using (db_fypEntities db = new db_fypEntities())
{
    var user = db.tb_student.Where(u => u.s_id == id).FirstOrDefault();

    if (user != null)
    {
        if (ModelState.IsValid)
        {
            if (upload != null && upload.ContentLength > 0)
            {
                var fileName = Path.GetFileName(upload.FileName);
                var path = Path.Combine(Server.MapPath("~/SVagreement/"), fileName);
                upload.SaveAs(path);

                user.s_SVagreement = upload.FileName;

                db.SaveChanges();
                TempData["success"] = "Agreement uploaded successfully!";
                return RedirectToAction("Index");
            }
            else
            {
                TempData["fail"] = "Please choose an agreement to upload. Only .pdf is accepted!";
                return View();
            }
        }
    }
}

```

To do so, an agreement action result must be created and save the PDF file into the SVagreement folder, the s_SVagreement in tb_student will keep the filename. Agreement View will be created so that the supervisor can upload the agreement in PDF form.

Committee

View All Students once they select supervisor :

The committee can manage all students and manage all proposals. Therefore, they can view students and the preferred supervisor once students select supervisors.

Edit:

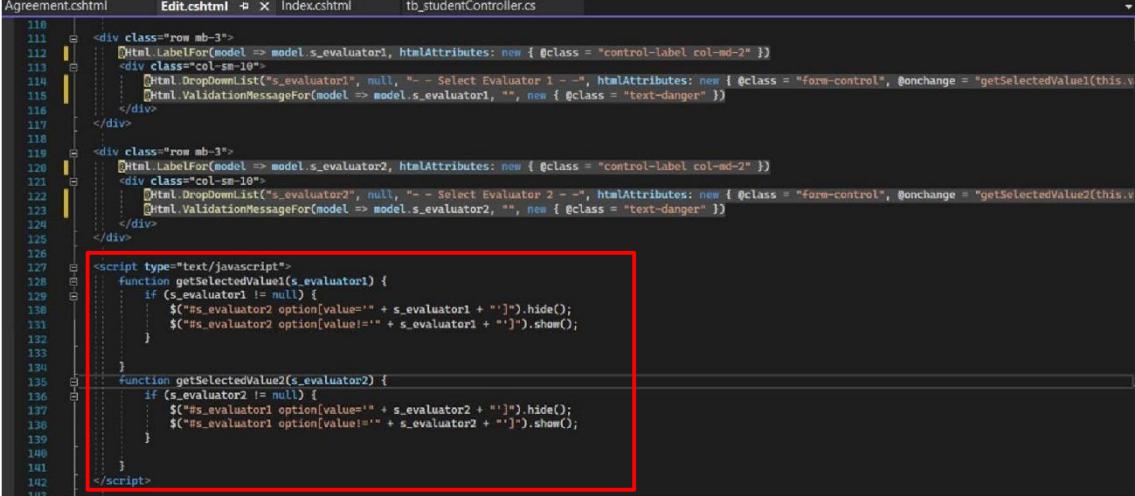
The committee will assign evaluators to students based on project type and the supervisor should not evaluate their own students' proposals. The selectList of evaluator 1 and evaluator 2 will not include the chosen supervisor, admin and students. Only lecturers who have the same domain with the project type of students will be displayed.

```

ViewBag.s_status_SV = new SelectList(db.tb_status.Where(u => u.st_id >= 1 && u.st_id <= 4), "st_id", "st_desc", tb_student.s_status_SV);
ViewBag.s_SV = new SelectList(db.tb_user, "u_id", "u_name", tb_student.s_SV);
ViewBag.s_user = new SelectList(db.tb_user, "u_id", "u_name", tb_student.s_user);
ViewBag.s_evaluator1 = new SelectList(db.tb_user.Where(u=>u.u_id != sv && u.u_type!=1 && u.u_type != 2 && u.u_domain==domain), "u_id", "u_name", tb_student.s_evaluator1);
ViewBag.s_evaluator2 = new SelectList(db.tb_user.Where(u=>u.u_id != sv && u.u_type != 1 && u.u_type != 2 && u.u_domain == domain), "u_id", "u_name", tb_student.s_evaluator2);
return View(tb_student);
}

```

In the edit view, javascript is used to prevent the committee from selecting the same person for evaluator 1 and evaluator 2. If one value is selected for evaluator 1, the selected value will be hidden from evaluator 2 and vice versa.



```
110 <div class="row mb-3">
111     <div class="col-sm-10">
112         @Html.LabelFor(model => model.s_evaluator1, htmlAttributes: new { @class = "control-label col-md-2" })
113         <div class="col-sm-10">
114             @Html.DropDownList("s_evaluator1", null, "-- Select Evaluator 1 --", htmlAttributes: new { @class = "form-control", @onchange = "getSelectedValue(this.value)" })
115             @Html.ValidationMessageFor(model => model.s_evaluator1, "", new { @class = "text-danger" })
116         </div>
117     </div>
118 
119     <div class="row mb-3">
120         @Html.LabelFor(model => model.s_evaluator2, htmlAttributes: new { @class = "control-label col-md-2" })
121         <div class="col-sm-10">
122             @Html.DropDownList("s_evaluator2", null, "-- Select Evaluator 2 --", htmlAttributes: new { @class = "form-control", @onchange = "getSelectedValue2(this.value)" })
123             @Html.ValidationMessageFor(model => model.s_evaluator2, "", new { @class = "text-danger" })
124         </div>
125     </div>
126 
127     <script type="text/javascript">
128         function getSelectedValue1(s_evaluator1) {
129             if (s_evaluator1 != null) {
130                 $("#s_evaluator2 option[value='" + s_evaluator1 + "']").hide();
131                 $("#s_evaluator2 option[value!=" + s_evaluator1 + "']").show();
132             }
133         }
134 
135         function getSelectedValue2(s_evaluator2) {
136             if (s_evaluator2 != null) {
137                 $("#s_evaluator1 option[value='" + s_evaluator2 + "']").hide();
138                 $("#s_evaluator1 option[value!=" + s_evaluator2 + "']").show();
139             }
140         }
141     </script>
142 
```

Delete:

The committee can delete all students and the PIC for students.

3.9 Proposal Module

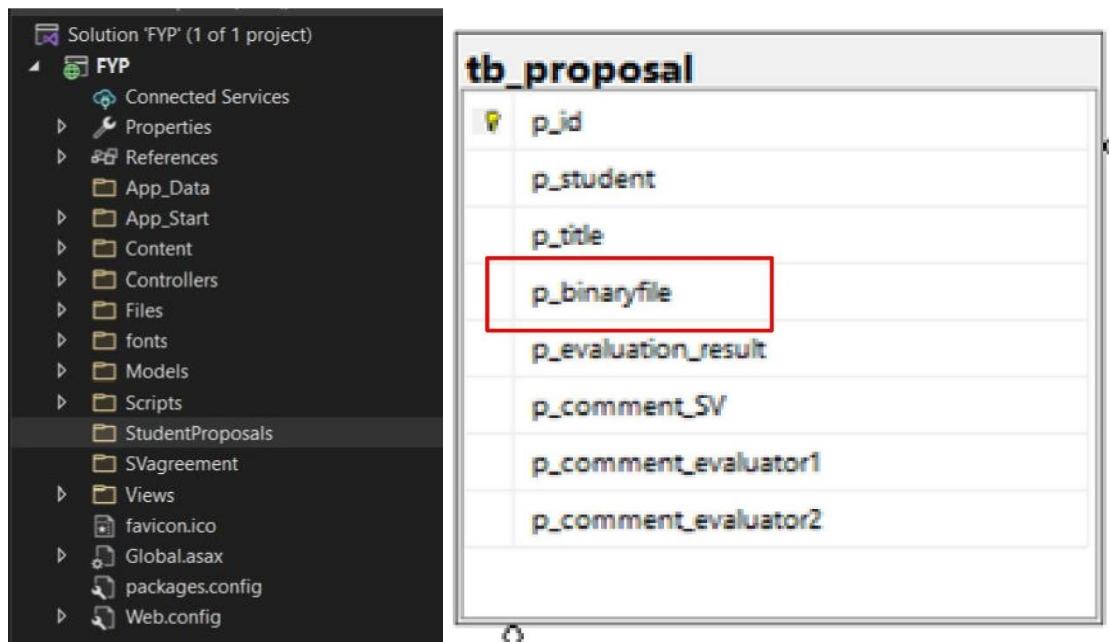
Role access:

- **Student** manage proposal, able to resubmit proposal
- **Committee** manage proposals
- **Lecturer (PIC in the proposal: Supervisor and Evaluators)** manage own students' proposals

The proposal module will store the information of student id, proposal title, proposal pdf filename, evaluation result, supervisor comments, evaluator 1 comment and evaluator 2 comment. The action result index will be filtered by academic session and semester of students in such a way that lecturer (Supervisor/Evaluators) can only view their own students' proposals based on academic session and semester whereas the committee can see all students' proposals who have submitted. Before that, students have to submit the proposal and fill in all needed information in the creation process.

3.9.1 Add a folder for All Students Proposals

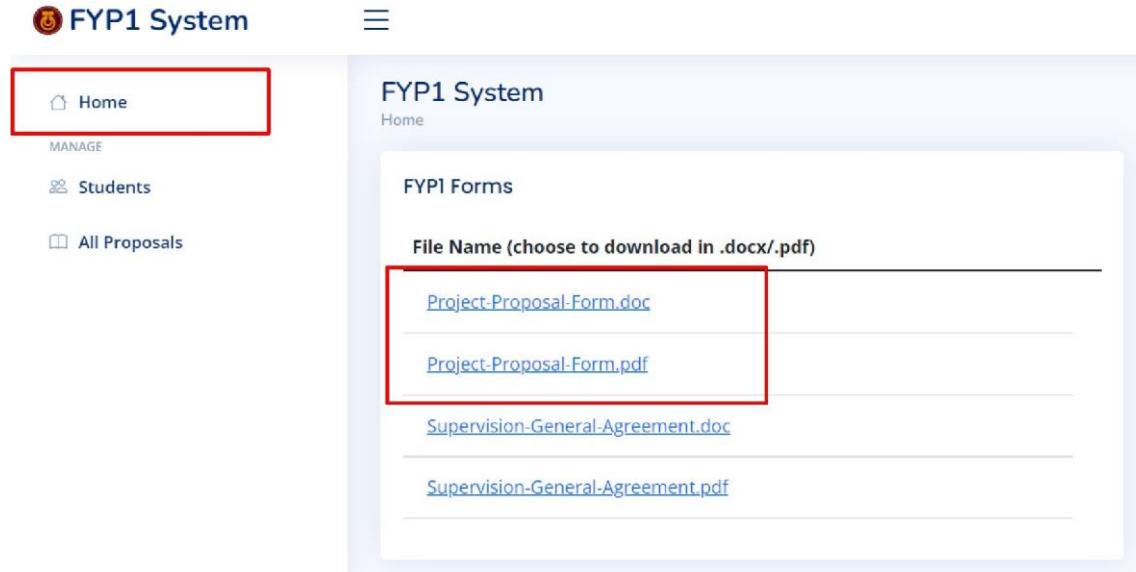
In this module, students have to upload the proposals after the selected supervisor is approved. Therefore, a new folder name named “StudentProposals” is created to store all of the uploaded proposals whereas the p_binaryfile in the tb_proposal is used to store the filename of the uploaded proposal PDF file.



tb_proposal

p_id
p_student
p_title
p_binaryfile
p_evaluation_result
p_comment_SV
p_comment_evaluator1
p_comment_evaluator2

The student can download the project proposal form in doc or pdf form on the home page, fill in all needed information, place signature and upload back to the system.



3.9.2 Create Role-Based Views

Student

Create (submit) proposal:

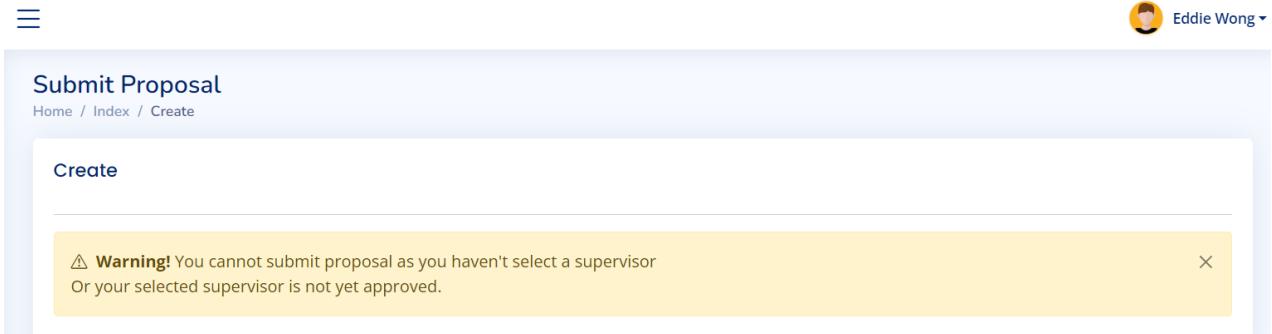
Students are the ones who submit the proposal, they have to enter the project title when submitting. The students are allowed to submit proposals once they have selected a supervisor and the status of selected supervisor is approved, evaluators 1 and 2 are assigned. The uploaded proposal will be saved into the StudentProposals folder while the filename will be saved into the p_binaryfile attribute.

```

207     public ActionResult Create(tb_proposal tb_proposal, HttpPostedFileBase upload)
208     {
209         if (ModelState.IsValid)
210         {
211             try
212             {
213                 if (upload != null && upload.ContentLength > 0)
214                 {
215                     var fileName = Path.GetFileName(upload.FileName);
216                     var path = Path.Combine(Server.MapPath("~/StudentProposals/"), fileName);
217                     upload.SaveAs(path);
218                 }
219             }
220             db.tb_proposal.Add(new tb_proposal
221             {
222                 p_student = tb_proposal.p_student,
223                 p_title = tb_proposal.p_title,
224                 p_evaluation_result = tb_proposal.p_evaluation_result,
225                 p_comment_SV = tb_proposal.p_comment_SV,
226                 p_comment_evaluator1 = tb_proposal.p_comment_evaluator1,
227                 p_comment_evaluator2 = tb_proposal.p_comment_evaluator2,
228                 p_binaryfile = upload.FileName
229             });
230
231             db.SaveChanges();
232             TempData["success"] = "Proposal information saved successfully";
233             return RedirectToAction("Index");
234         }
235         catch (Exception ex)
236     }

```

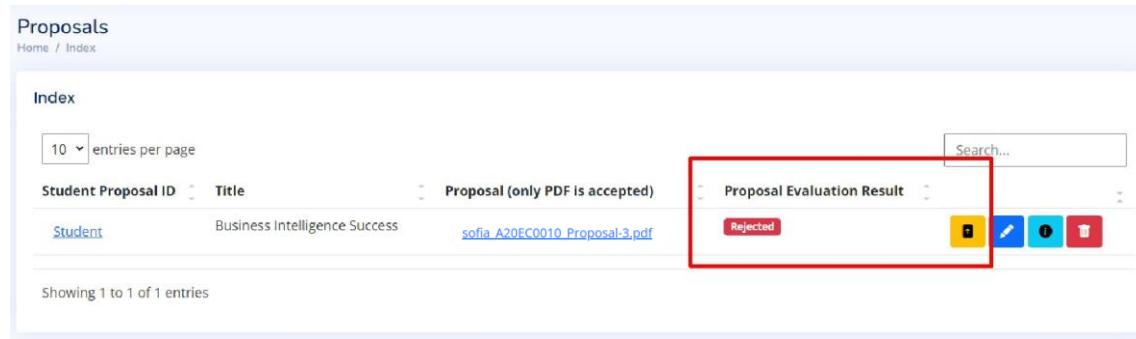
If the students have not selected a supervisor and the status of selected supervisor is not approved, students are not allowed to submit proposals as shown below.



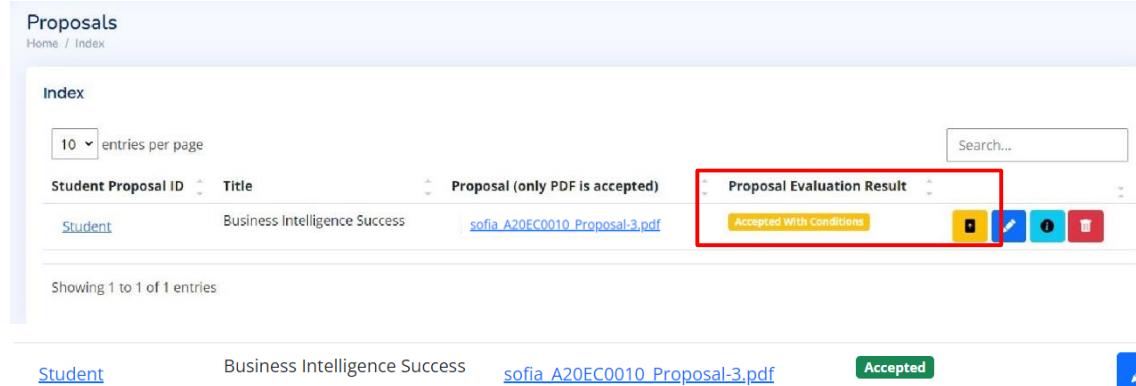
The screenshot shows a 'Submit Proposal' page with a 'Create' button. A yellow warning box displays the message: '⚠ Warning! You cannot submit proposal as you haven't select a supervisor Or your selected supervisor is not yet approved.' An 'X' icon is in the top right corner of the box.

Edit (Resubmit proposals):

Students are allowed to resubmit proposals if the status is “rejected” or “accepted with conditions”.



The screenshot shows a 'Proposals' index page with a table. The 'Proposal Evaluation Result' column for the first row is highlighted with a red box and contains the text 'Rejected'. The row details a proposal by a student titled 'Business Intelligence Success' with the file 'sofia_A20EC0010_Proposal-3.pdf'.



The screenshot shows a 'Proposals' index page with a table. The 'Proposal Evaluation Result' column for the first row is highlighted with a red box and contains the text 'Accepted With Conditions'. The row details a proposal by a student titled 'Business Intelligence Success' with the file 'sofia_A20EC0010_Proposal-3.pdf'.

View all details:

Students can view all details of proposals like student information, title, download the submitted proposal, comments from supervisor and two evaluators and proposal evaluation result. Students can view the PIC for their proposal by clicking the “Student” link.

Details	
Student Proposal ID	Student
Title	Business Intelligence Success
Proposal (only PDF is accepted)	sofia_A20FC0010_Proposal-3.pdf
Supervisor Comment	Good and complete, under guidance.
Evaluator 1 Comment	Relevant, mature
Evaluator 2 Comment	mature and good
Proposal Evaluation Result	Accepted

After clicking the Student Link, all of the people in charge in the proposal will be shown.

PIC in Proposal	
Home / Index / Details	
Details	
Academic Session	2021/2022
Semester	1
Student	Sofia Ahmad
Supervisor	Alexander
Supervisor Agreement	alex_sofia.pdf
Supervisor Approval Status	Approved
Evaluator 1	Iqbal Hakim
Evaluator 2	Lim Mei Ying
Proposal Details	Proposal Details

Delete:

Students can delete the submitted proposals.

Lecturer (PIC in the proposal: Supervisor and Evaluators)

View Own Students’ Proposal (Filter by semester and academic session):

By editing the codes inside the Index, the PIC in the proposal is able to view its own students' proposals. If the Session id is corresponded to the id of PIC for students proposals, related data will be shown.

```
//PIC manage own students' proposals
else
{
    var tb_proposal = db.tb_proposal
    .Include(t => t.tb_status)
    .Include(t => t.tb_student)
    .Where(t => (t.tb_student.s_sv == id1 || t.tb_student.s_evaluator1 == id1 || t.tb_student.s_evaluator2 == id1) && t.tb_student.s_semester == s_semester && t.tb_student.s_academic_session == s_academic_session);
    return View(tb_proposal.ToList());
}
```

The supervisor is able to view the evaluation result after students submit the proposals. Evaluators and supervisors can download all related proposals.

Clicking the blue info button, the supervisor can view the comments from evaluators.

Edit:

The supervisor must leave a comment before the evaluator assessment. Otherwise, the evaluation result field is unavailable.

Edit

Comment Student's Proposal

Student Involved	8
Title	Analytics on Data
Evaluator 2 Comment	
Proposal Evaluation Result	-- Available after Supervisor leave comments --

Save **Cancel**

After the supervisor leaves comments, evaluation result field is available and evaluators can choose status for evaluation result.

FYP1 System ☰ Iqbal Hakim

Form Layouts Home / Index / Edit

Edit

Comment Student's Proposal

Student Involved	9
Title	Business Intelligence Success
Evaluator 1 Comment	Relevant, mature
Proposal Evaluation Result	... Select evaluation result ... <ul style="list-style-type: none"> ... Select evaluation result ... rejected accepted accepted with conditions

Save **Cancel**

© Designed by Ham Jing Yi, Data Engineering (UTM)

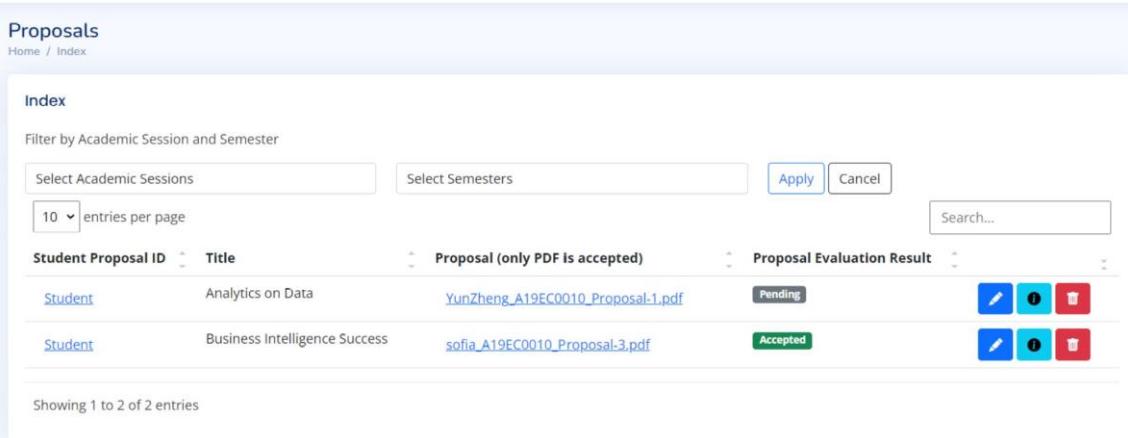
I do face a problem here to save the filename after somebody edit the form. However, I manage to solve it by using a static variable to save the filename inside the variable and a hidden field to retrieve and save the data according to different user roles. ViewData is used to display and hide the comments field of evaluators and supervisors in the edit view according to the session role.

Delete: PIC can delete the selected row of data.

Committee

View (Filter by semester and academic session):

The committee is able to view all students' proposals. Therefore, they can download all proposals once students have submitted them. The committee can filter all proposals by academic session and semesters.



Proposals

Home / Index

Index

Filter by Academic Session and Semester

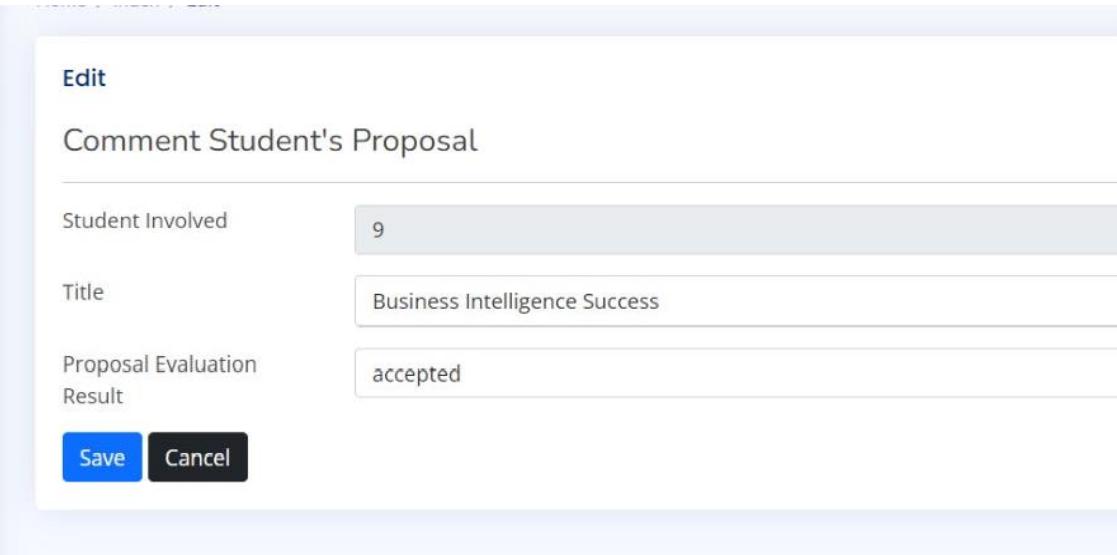
Select Academic Sessions Select Semesters Apply Cancel

10 entries per page Search...

Student Proposal ID	Title	Proposal (only PDF is accepted)	Proposal Evaluation Result
Student	Analytics on Data	YunZheng_A19EC0010_Proposal-1.pdf	Pending
Student	Business Intelligence Success	sofia_A19EC0010_Proposal-3.pdf	Accepted

Showing 1 to 2 of 2 entries

Edit: Committee can edit the proposal evaluation result of all proposals.



Edit

Comment Student's Proposal

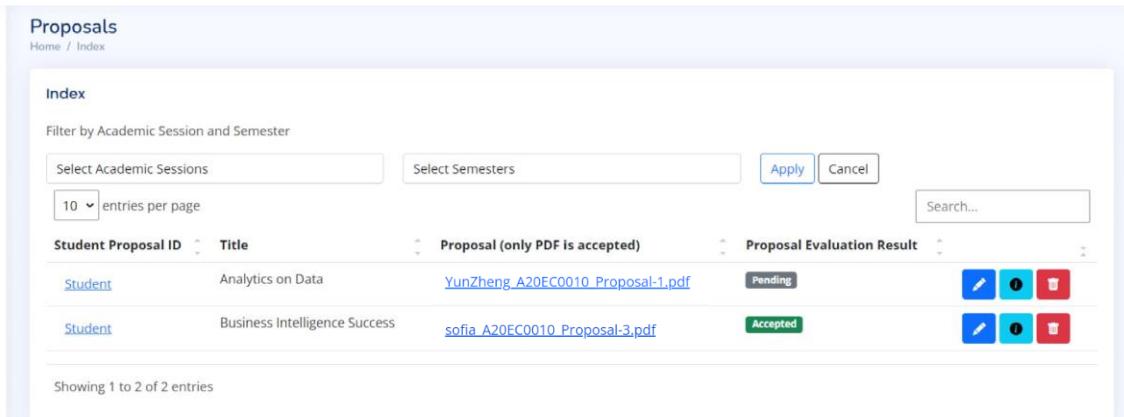
Student Involved 9

Title Business Intelligence Success

Proposal Evaluation Result accepted

Save Cancel

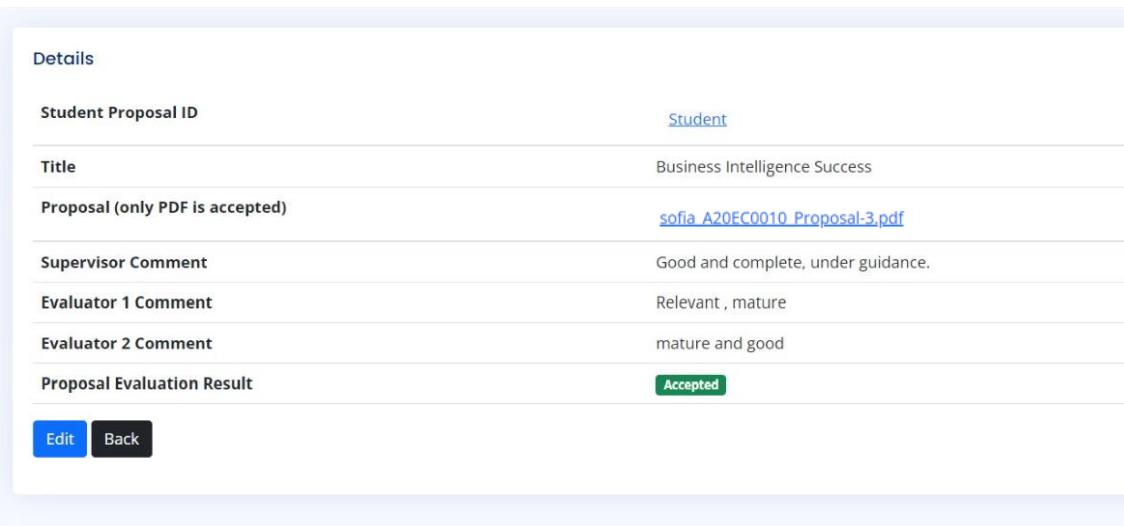
The committee is able to view the evaluation result, and all comments after students submit the proposals. Committee can download all related proposals.



The screenshot shows a table of proposals with the following data:

Student Proposal ID	Title	Proposal (only PDF is accepted)	Proposal Evaluation Result
Student	Analytics on Data	YunZheng_A20EC0010_Proposal-1.pdf	Pending
Student	Business Intelligence Success	sofia_A20EC0010_Proposal-3.pdf	Accepted

Clicking the blue info button, the committee can view the comments from supervisors and evaluators.

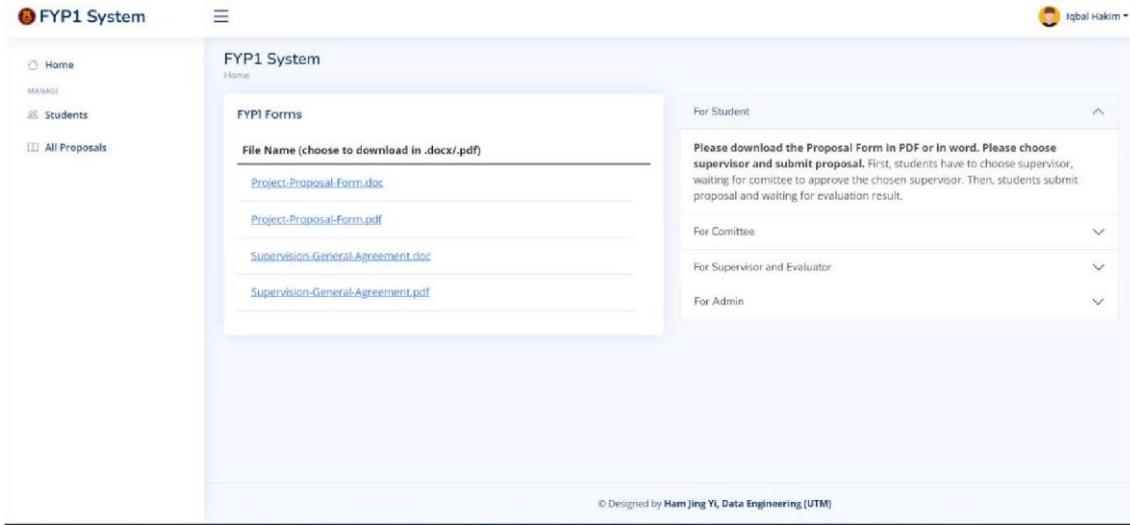


The screenshot shows the following proposal details:

Student Proposal ID	Student
Title	Business Intelligence Success
Proposal (only PDF is accepted)	sofia_A20EC0010_Proposal-3.pdf
Supervisor Comment	Good and complete, under guidance.
Evaluator 1 Comment	Relevant, mature
Evaluator 2 Comment	mature and good
Proposal Evaluation Result	Accepted

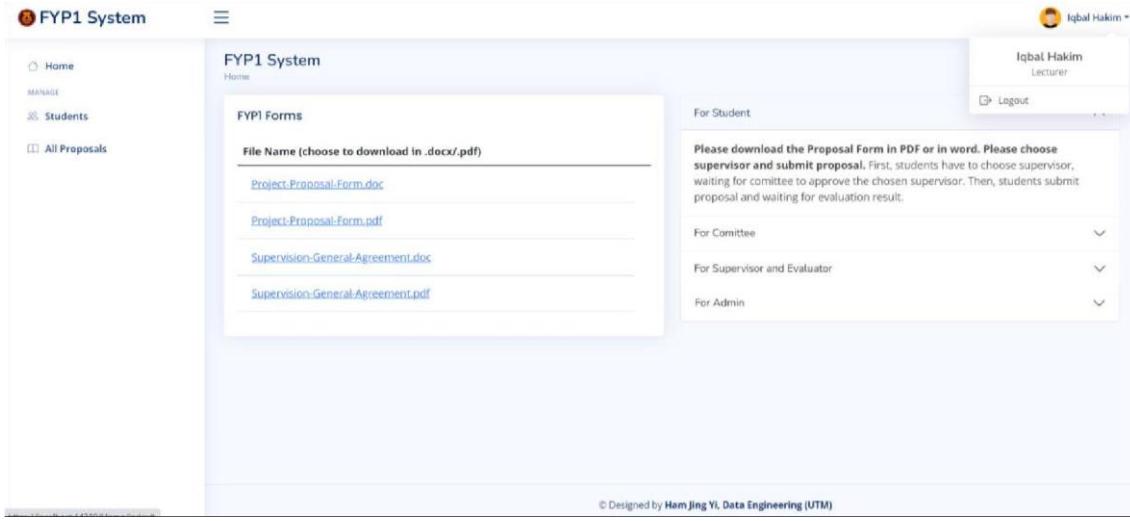
Delete: committee can delete the uploaded proposals.

3.10 Customise the Home Index, Navigation Sidebar and All Modules



The screenshot shows the FYP1 System home page. On the left, a navigation sidebar lists 'Home', 'MANAGE', 'Students', and 'All Proposals'. The main content area is titled 'FYP1 System' and 'Home'. It features a section for 'FYP1 Forms' with a list of download links: 'Project-Proposal-Form.doc', 'Project-Proposal-Form.pdf', 'Supervision-General-Agreement.doc', and 'Supervision-General-Agreement.pdf'. To the right, a sidebar titled 'For Student' contains a message: 'Please download the Proposal Form in PDF or in word. Please choose supervisor and submit proposal. First, students have to choose supervisor, waiting for committee to approve the chosen supervisor. Then, students submit proposal and waiting for evaluation result.' Below this are sections for 'For Committee', 'For Supervisor and Evaluator', and 'For Admin', each with a dropdown arrow. At the bottom right of the main content area, there is a copyright notice: '© Designed by Ham Jing Yi, Data Engineering (UTM)'.

The home page is designed in such a way that supervisor and students can download the needed form in doc or pdf. The navigation bar menu will be displayed differently according to session role. Button sizing, position, card body is added to all modules.



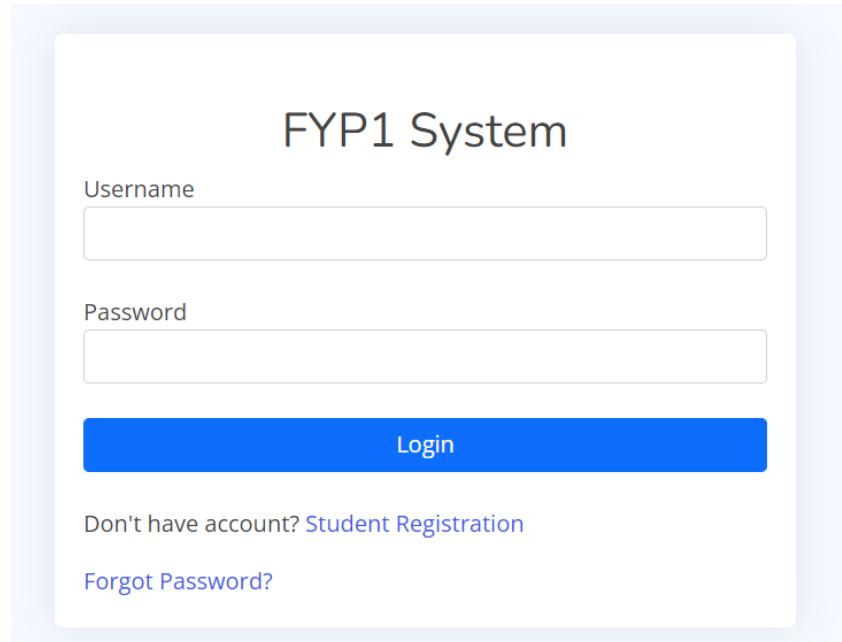
This screenshot is identical to the one above, showing the FYP1 System home page. The navigation sidebar, main content area with FYP1 Forms, and the sidebar for 'For Student' are all present. The key difference is the user profile 'Iqbal Hakim' is now visible in the top right corner, indicating the current logged-in user.

On the top right, the full name and the user type of the login user will be shown.

4.0 System interface for all users

General

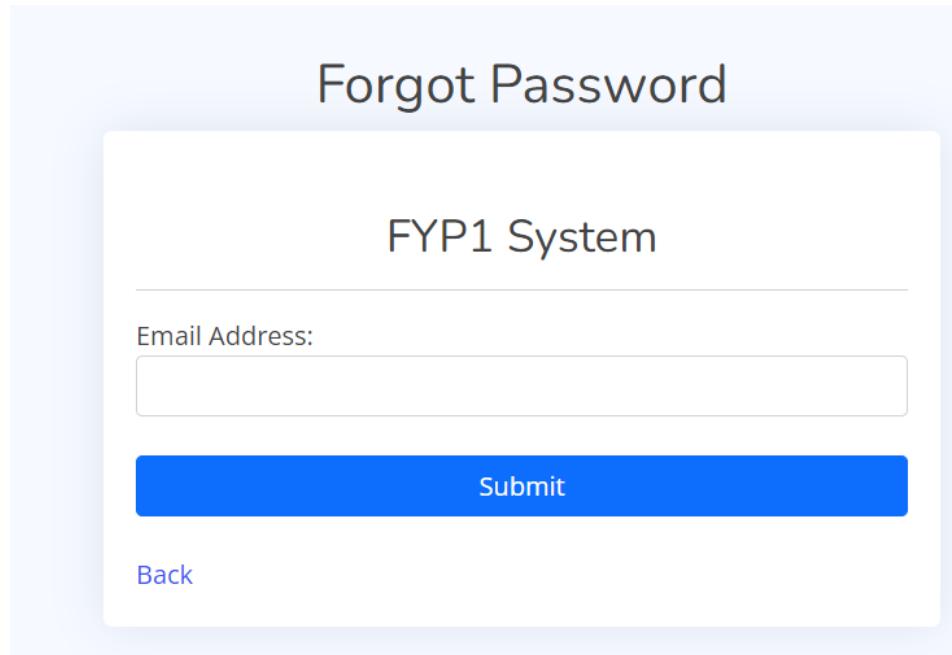
Login Page



The image shows the login interface for the FYP1 System. It features a light gray header with the text "FYP1 System" in a dark gray font. Below the header is a "Username" label with a corresponding input field. Underneath is a "Password" label with a corresponding input field. A large blue "Login" button is centered below the input fields. At the bottom of the interface, there are two links: "Don't have account? [Student Registration](#)" and "Forgot Password?".

Figure 4.1: Interface for <Login>

Forgot Password Page



The image shows the forgot password interface for the FYP1 System. It has a light gray header with the text "Forgot Password" in a dark gray font. Below the header is a "FYP1 System" logo. A "Email Address:" label is followed by an input field. A large blue "Submit" button is centered below the input field. At the bottom of the interface, there is a "Back" link.

Figure 4.2: Interface for <Forgot Password>

Reset Password

FYP1 System

New Password

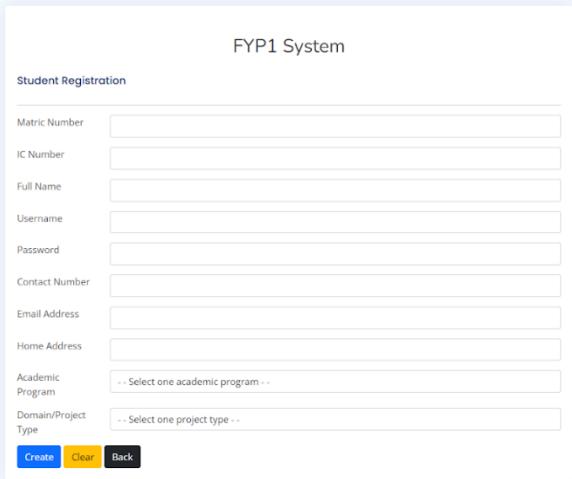
Confirm Password

[Submit](#)

[Login](#)

Figure 4.3: Interface for <Reset Password>

Student Registration



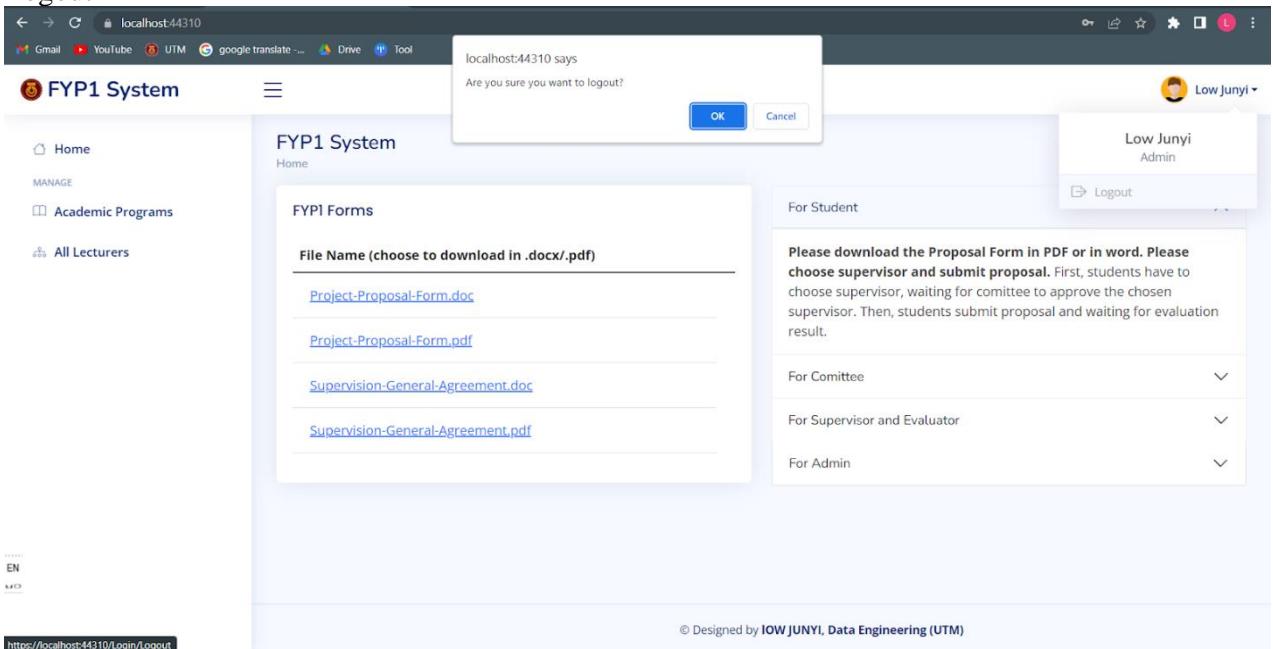
The screenshot shows the 'Student Registration' form within the 'FYP1 System'. The form contains the following fields:

- Matrix Number
- IC Number
- Full Name
- Username
- Password
- Contact Number
- Email Address
- Home Address
- Academic Program (dropdown: -- Select one academic program --)
- Domain/Project Type (dropdown: -- Select one project type --)

At the bottom of the form are three buttons: 'Create' (blue), 'Clear' (yellow), and 'Back' (black).

Figure 4.4: Interface for <Student Registration>

Logout

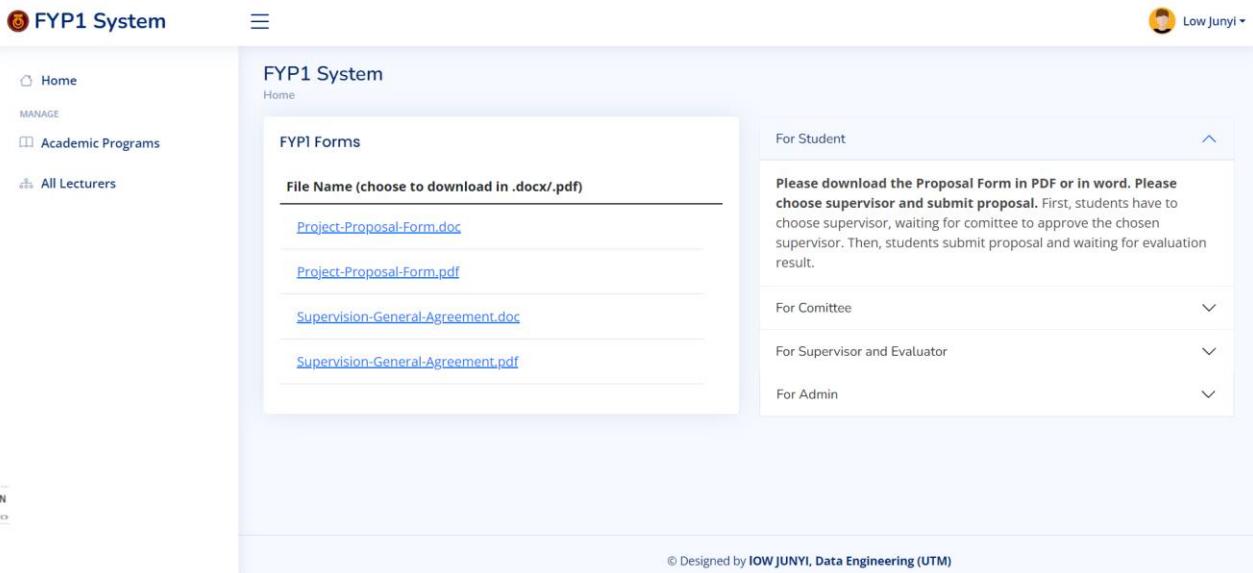


The screenshot shows a browser window for the 'FYP1 System' on 'localhost:44310'. A modal dialog box is centered, asking 'localhost:44310 says Are you sure you want to logout?'. The 'OK' button is highlighted in blue, and the 'Cancel' button is in white. The browser's address bar shows the URL 'https://localhost:44310/Login/Logout'. The page content includes a sidebar with 'Home', 'MANAGE', 'Academic Programs', and 'All Lecturers'. The main content area is titled 'FYP1 Forms' and lists download links for 'Project-Proposal-Form.doc', 'Project-Proposal-Form.pdf', 'Supervision-General-Agreement.doc', and 'Supervision-General-Agreement.pdf'. To the right, a sidebar for 'Low Junyi' (Admin) shows links for 'Logout', 'For Student', 'For Committee', 'For Supervisor and Evaluator', and 'For Admin'. A note at the bottom of the sidebar reads: 'Please download the Proposal Form in PDF or in word. Please choose supervisor and submit proposal. First, students have to choose supervisor, waiting for committee to approve the chosen supervisor. Then, students submit proposal and waiting for evaluation result.'

Figure 4.5: Interface for <Logout>

4.2.1 Admin

Home Page



FYP1 System

Home

MANAGE

Academic Programs

All Lecturers

FYP1 System

Home

FYP1 Forms

File Name (choose to download in .docx/.pdf)

[Project-Proposal-Form.doc](#)

[Project-Proposal-Form.pdf](#)

[Supervision-General-Agreement.doc](#)

[Supervision-General-Agreement.pdf](#)

For Student

Please download the Proposal Form in PDF or in word. Please choose supervisor and submit proposal. First, students have to choose supervisor, waiting for committee to approve the chosen supervisor. Then, students submit proposal and waiting for evaluation result.

For Committee

For Supervisor and Evaluator

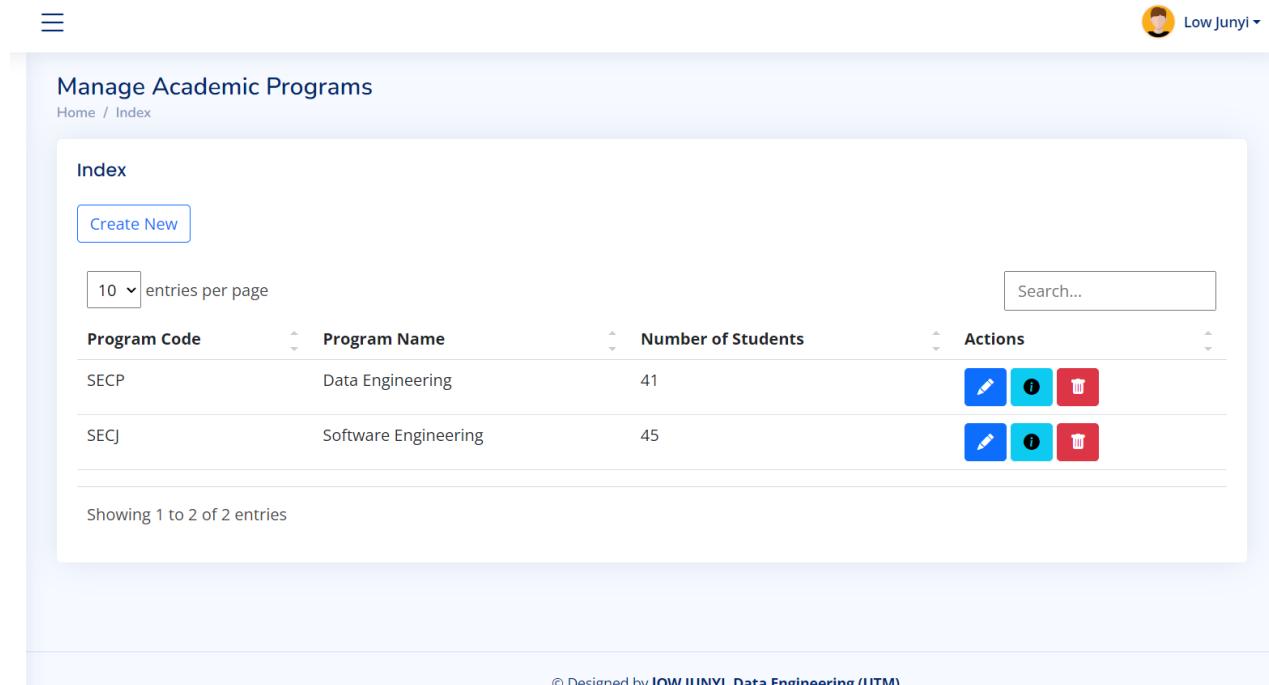
For Admin

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.6: Interface for <Admin's Home Page>

Menu 1: Academic Program

Read



Manage Academic Programs

Home / Index

Index

Create New

10 entries per page

Search...

Program Code	Program Name	Number of Students	Actions
SECP	Data Engineering	41	
SECJ	Software Engineering	45	

Showing 1 to 2 of 2 entries

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.7: Interface for <Read Academic Programs>

Create



Manage Academic Programs

Home / Index / Create

Create

New Academic Program

Program Code

Program Name

Number of Students

Create **Clear** **Back**

© Designed by **LOW JUNYI, Data Engineering (UTM)**

Figure 4.8: Interface for <Create Academic Programs>

Update



Form Layouts

Home / Index / Edit

Edit

Academic Program

Program Code

Program Name

Number of Students

Save **Back**

© Designed by **LOW JUNYI, Data Engineering (UTM)**

Figure 4.9: Interface for <Update Academic Programs>

Details

Manage Academic Programs

Home / Index / Details

Details

Program Code	SECJ
Program Name	Software Engineering
Number of Students	45

[Edit](#) [Back](#) [Lecturer List For This Academic Program](#)

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.10: Interface for <Read Details of Academic Programs>

Matric Number	Full Name	Program Name	Domain/Project Type	User Type	Actions
A20EC0071	Low Junyi	Data Engineering	Not yet assigned	admin	
12344	Alexander	Software Engineering	Development	lecturer	
12341	Siti Hajar	Data Engineering	Not yet assigned	committee	
34122	Iqbal Hakim	Data Engineering	Development	lecturer	
12122	Lim Mei Ying	Data Engineering	Development	lecturer	
23221	Farah Ann	Software Engineering	Research	lecturer	

Showing 1 to 6 of 6 entries

© Designed by IOW JUNYI, Data Engineering (UTM)

Add Committee

Remove Committee

Figure 4.11: Interface for <Remove/ Assign Committee Role>

Delete



Manage Academic Programs

Home / Index / Delete

Delete

Are you sure you want to delete this?

Program Code	SECJ
Program Name	Software Engineering
Number of Students	45

Delete **Back**

Figure 4.12: Interface for <Delete Academic Programs>

Menu 2: All Lecturers

Read


Low Junyi ▾

[Create New](#)

10 entries per page

Matric Number	Full Name	Program Name	Domain/Project Type	User Type	Actions
A20EC0071	Low Junyi	Data Engineering	Not yet assigned	admin	   
12344	Alexander	Software Engineering	Development	lecturer	   
12341	Siti Hajar	Data Engineering	Not yet assigned	committee	   
34122	Iqbal Hakim	Data Engineering	Development	lecturer	   
12122	Lim Mei Ying	Data Engineering	Development	lecturer	   
23221	Farah Ann	Software Engineering	Research	lecturer	   

Showing 1 to 6 of 6 entries

© Designed by **LOW JUNYI, Data Engineering (UTM)** 

Figure 4.13: Interface for <Read All Lecturers>

Create


Low Junyi ▾

Create

New User

IC Number	<input type="text"/>	Full Name	<input type="text"/>
Username	<input type="text"/>	Password	<input type="text"/>
Matric Number	<input type="text"/>	Email Address	<input type="text"/>
Contact Number	<input type="text"/>	Home Address	<input type="text"/>
Academic Program	<input type="text" value="-- Select one academic program --"/>		
User Type	<input type="text" value="-- Select one user type --"/>		
<input type="button" value="Create"/> <input type="button" value="Clear"/> <input type="button" value="Back"/>			

© Designed by **LOW JUNYI, Data Engineering (UTM)**

Figure 4.14: Interface for <Create New Lecturers>

Update



Low Junyi ▾

Edit

User

IC Number	010308141213	Full Name	Low Junyi
Username	lowjunyi	Email Address	low.junyi2001@gmail.com
Matric Number	A20EC0071	Home Address	KTDI
Contact Number	0182858291		
Academic Program	Data Engineering		
User Type	admin		

Save **Cancel**

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.15: Interface for <Update Lecturers>

Details



Low Junyi ▾

Manage Lecturers

Home / Index / Details

Details

IC Number	010308141213
Matric Number	A20EC0071
Full Name	Low Junyi
Username	lowjunyi
Contact Number	0182858291
Email Address	low.junyi2001@gmail.com
Home Address	KTDI
Program Code	SECP
Domain/Project Type	Not yet assigned
User Type	admin

Edit **Back**

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.16: Interface for <Read Details of Lecturers>

Delete



Low Junyi ▾

Delete

Are you sure you want to delete this?

IC Number	010308141213
Matric Number	A20EC0071
Full Name	Low Junyi
Username	lowjunyi
Contact Number	0182858291
Email Address	low.junyi2001@gmail.com
Home Address	KTDI
Program Code	SECP
Domain/Project Type	Not yet assigned
User Type	admin

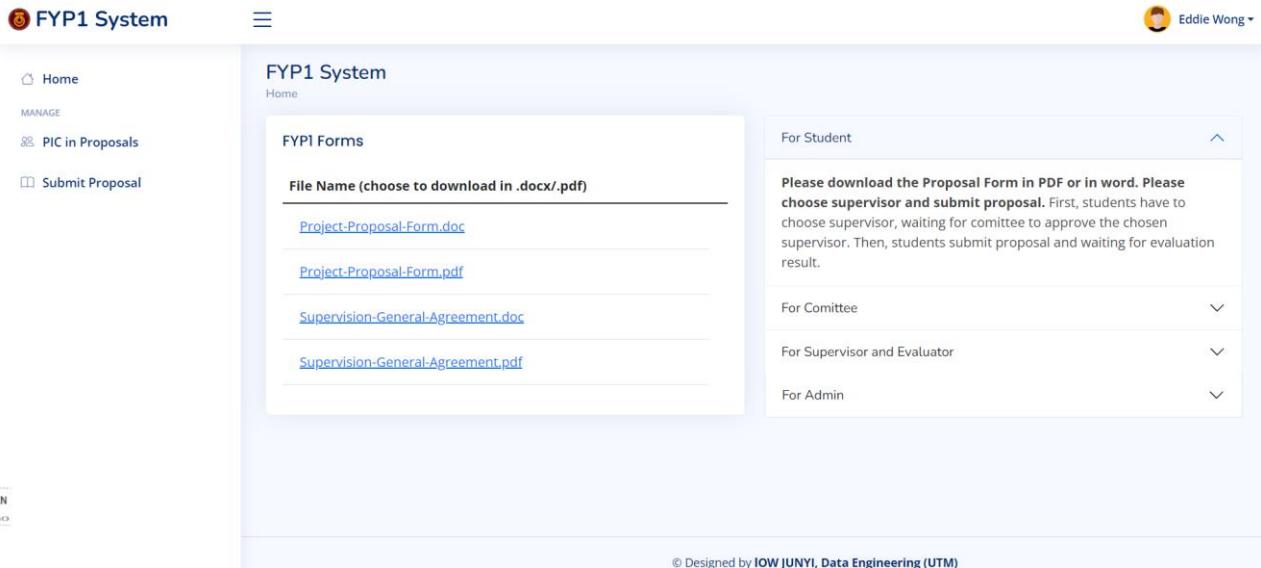
Delete **Back**

© Designed by **LOW JUNYI, Data Engineering (UTM)**

Figure 4.17: Interface for <Delete Lecturers>

4.2.2 Student

Home Page



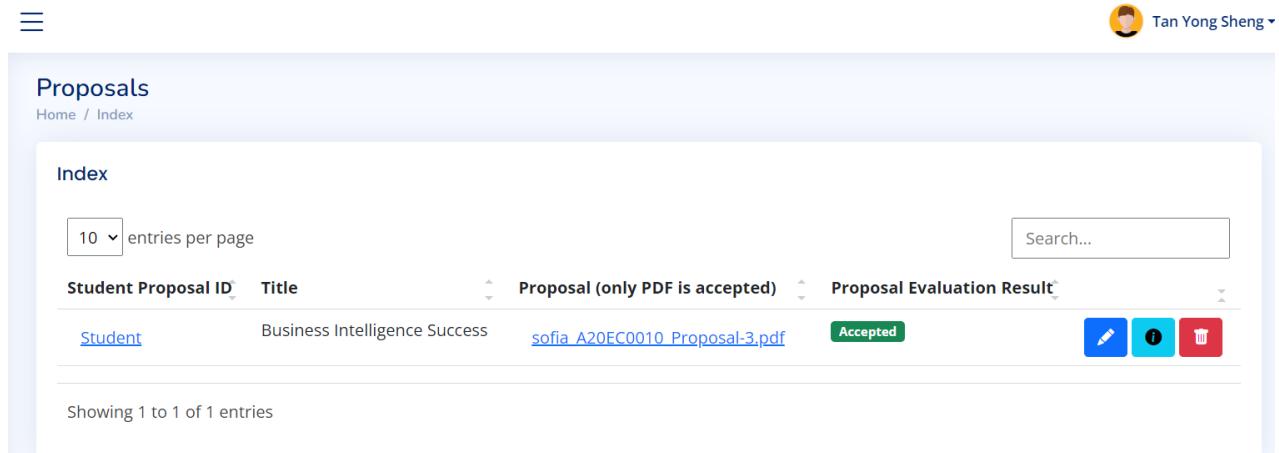
© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.18: Interface for <Student's Home Page>

Menu 1: PIC in proposals

Read

1. If Supervisor is approved



Student Proposal ID	Title	Proposal (only PDF is accepted)	Proposal Evaluation Result
Student	Business Intelligence Success	sofia_A20EC0010_Proposal-3.pdf	Accepted

Figure 4.19: Interface for <Read PIC in Proposals>

2. If Supervisor is not approved

Proposals

Home / Index

Index

10 entries per page

Search...

Student Proposal ID	Title	Proposal (only PDF is accepted)	Proposal Evaluation Result
abcd	AD_IndividualTest_2023.pdf	Pending	

Showing 1 to 1 of 1 entries

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.20: Interface for <Read PIC in Proposals>

Create

PIC in Proposal

Home / Index / Create

Create

Select Available Supervisor

Student Name: Eddie Wong

Academic Session (eg. 2021/2022):

Semester (eg. 1):

Supervisor Name: -- Select preferred supervisor --

Create

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.20: Interface for <Select Supervisor in Proposals>

Update



Tan Yong Sheng ▾

Select Preferred Supervisor

[Home](#) / [Index](#) / [Edit](#)

Edit

Assign Evaluators

Student Name

Tan Yong Sheng

Semester (eg. 1)

1

Academic Session (eg. 2022/2023)

2022/2023

Supervisor Name

Alexander

Evaluator 1 Name

Iqbal Hakim

Evaluator 2 Name

Lim Mei Ying

Save

Back

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.21: Interface for <Update Selected Supervisor Information in Proposals>

Details



Tan Yong Sheng ▾

Details

Academic Session

2022/2023

Semester

1

Student

[Tan Yong Sheng](#)

Supervisor

[Alexander](#)

Supervisor Agreement

Not Available

Supervisor Approval Status

Pending

Evaluator 1

[Iqbal Hakim](#)

Evaluator 2

[Lim Mei Ying](#)

Proposal Details

[Proposal Details](#)

Edit

Back

Figure 4.22: Interface for <Details of PIC for Proposals>

Delete



Tan Yong Sheng ▾

PIC in Proposal

Home / Index / Delete

Delete

Are you sure you want to delete this?

Academic Session (eg. 2022/2023)	2022/2023
Semester (eg. 1)	1
Student	Tan Yong Sheng
Supervisor	Alexander
Supervisor Agreement	Not Available
Supervisor Approval Status	Pending
Evaluator 1	Iqbal Hakim
Evaluator 2	Lim Mei Ying

Delete **Back**

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.23: Interface for <Delete of PIC for Proposals if status is pending or rejected>

Menu 2: Submit Proposals

Read



Tan Yong Sheng ▾

Proposals

Home / Index

Index

Student Proposal ID	Title	Proposal (only PDF is accepted)	Proposal Evaluation Result	
Student	Business Intelligence Success	sofia_A20EC0010_Proposal-3.pdf	Accepted	

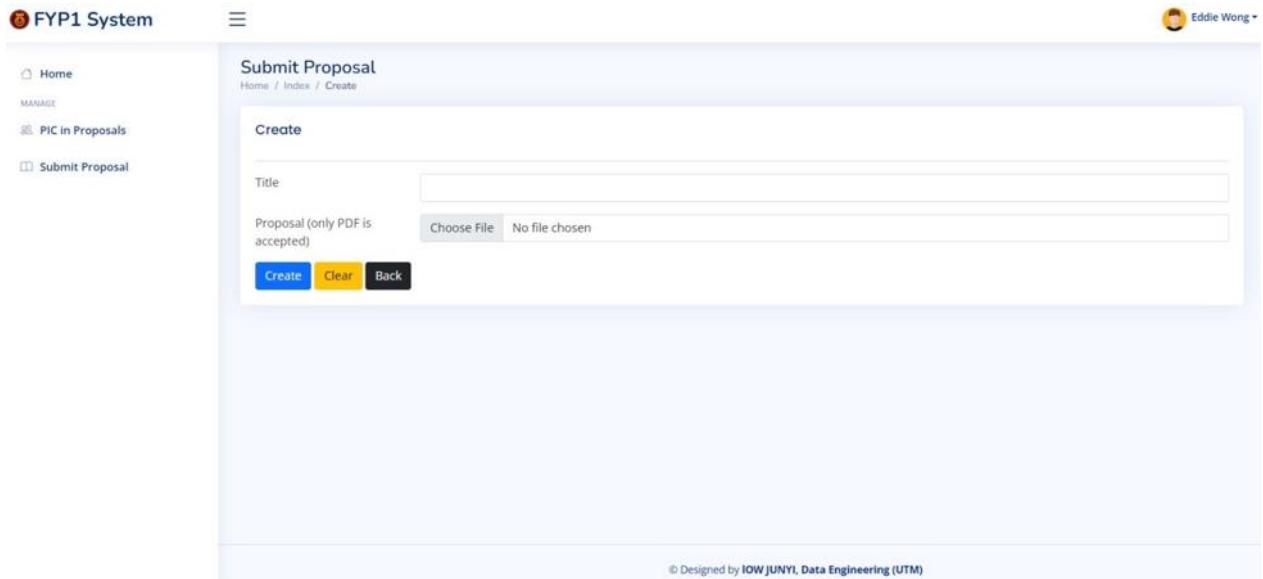
Showing 1 to 1 of 1 entries

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.24: Interface for <Read Submitted Proposals>

Create

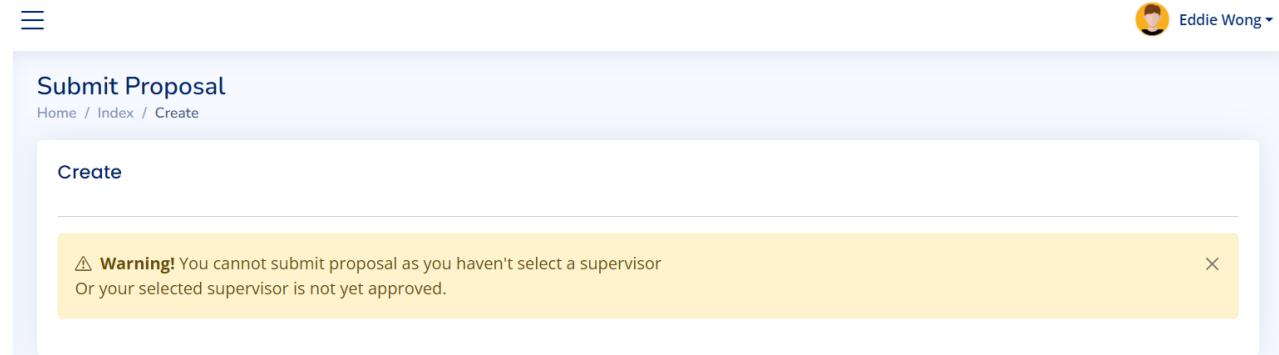
1. If Supervisor is approved



The screenshot shows a web application interface for submitting a proposal. The top navigation bar includes a logo for 'FYP1 System', a user profile for 'Eddie Wong', and a 'MANAGE' menu. The main content area is titled 'Submit Proposal' and 'Create'. It features a 'Title' input field, a 'Proposal' file input field with a 'Choose File' button and a message 'No file chosen', and three buttons: 'Create' (blue), 'Clear' (yellow), and 'Back' (black). The footer of the page credits '© Designed by IOW JUNYI, Data Engineering (UTM)'.

Figure 4.25: Interface for <Submit Proposals>

2. If Supervisor is not approved



The screenshot shows the same 'Submit Proposal' interface as Figure 4.25, but with a yellow warning box displayed. The box contains the text: '⚠ Warning! You cannot submit proposal as you haven't select a supervisor Or your selected supervisor is not yet approved.' There is a close 'X' button in the top right corner of the box.

Figure 4.26: Interface for <Submit Proposals>

Update



Tan Yong Sheng ▾

Proposals
Home / Index / Edit

Edit

Comment Student's Proposal

Student Involved	9
Title	Business Intelligence Success

Save **Cancel**

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.27: Interface for <Update Proposals Title>

Details



Tan Yong Sheng ▾

Proposals
Home / Index / Details

Details

Student Proposal ID	<u>Student</u>
Title	Business Intelligence Success
Proposal (only PDF is accepted)	sofia_A20EC0010_Proposal-3.pdf
Supervisor Comment	Good and complete, under guidance.
Evaluator 1 Comment	Relevant, mature
Evaluator 2 Comment	mature and good
Proposal Evaluation Result	Accepted

Edit **Back**

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.28: Interface for <Read Details Proposals>

Delete



Tan Yong Sheng ▾

Proposals

Home / Index / Delete

Delete

Are you sure you want to delete this?

Student Proposal ID	Student
Title	Business Intelligence Success
Proposal (only PDF is accepted)	sofia_A20EC0010_Proposal-3.pdf
Supervisor Comment	Good and complete, under guidance.
Evaluator 1 Comment	Relevant, mature
Evaluator 2 Comment	mature and good
Proposal Evaluation Result	Accepted

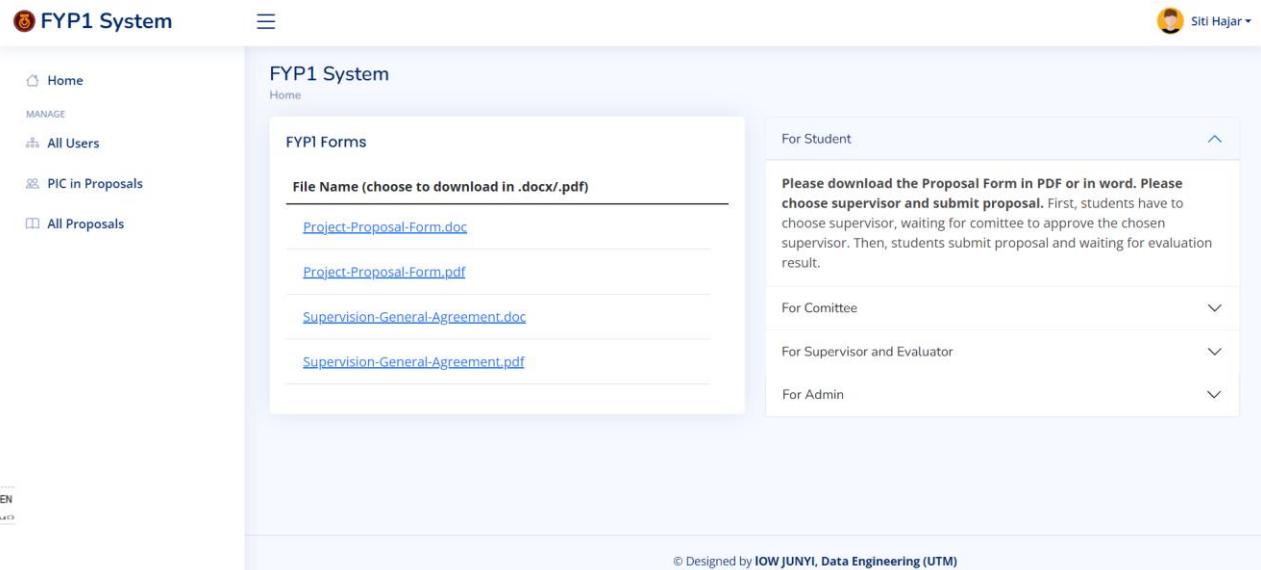
Delete **Back**

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.29: Interface for <Delete Proposals>

4.2.3 Committee

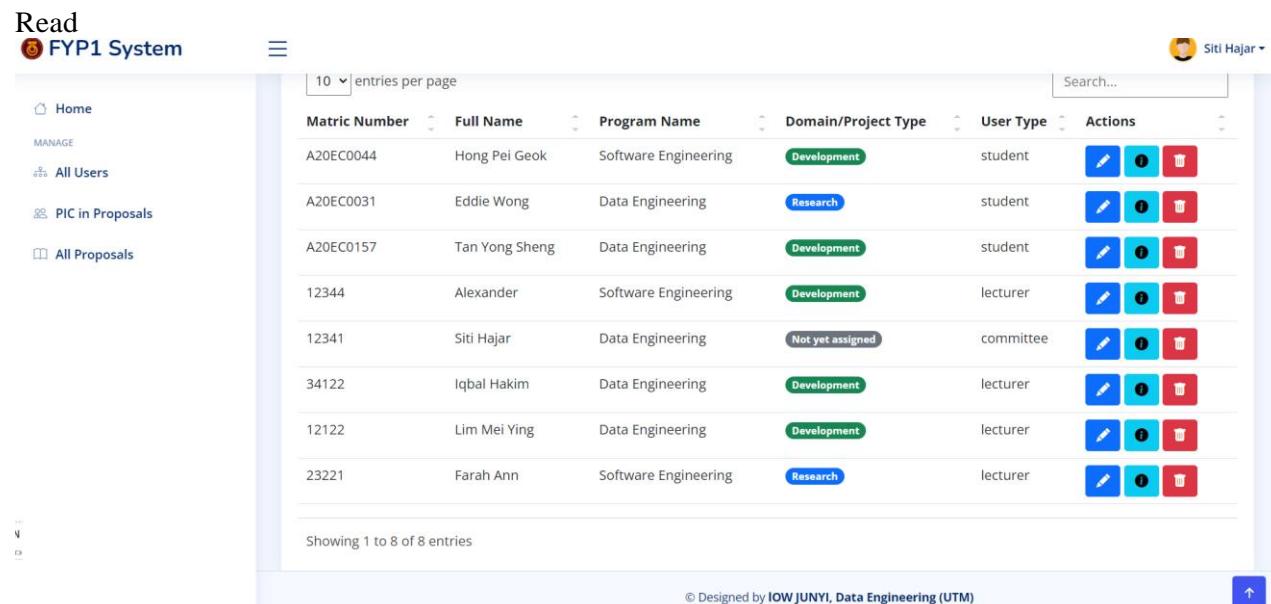
Home Page



The screenshot shows the FYP1 System Home Page. The left sidebar has links for Home, Manage (All Users, PIC in Proposals, All Proposals), and a language switch (EN). The main content area is titled 'FYP1 System' and 'Home'. It features a section for 'FYP1 Forms' with download links for 'Project-Proposal-Form.doc' and 'Project-Proposal-Form.pdf'. Below this are links for 'Supervision-General-Agreement.doc' and 'Supervision-General-Agreement.pdf'. To the right, there are sections for 'For Student', 'For Committee', 'For Supervisor and Evaluator', and 'For Admin'. Each section has a collapse/expand arrow. At the bottom, it says '© Designed by IOW JUNYI, Data Engineering (UTM)'.

Figure 4.30: Interface for <Committee's Home Page>

Menu 1: All Users

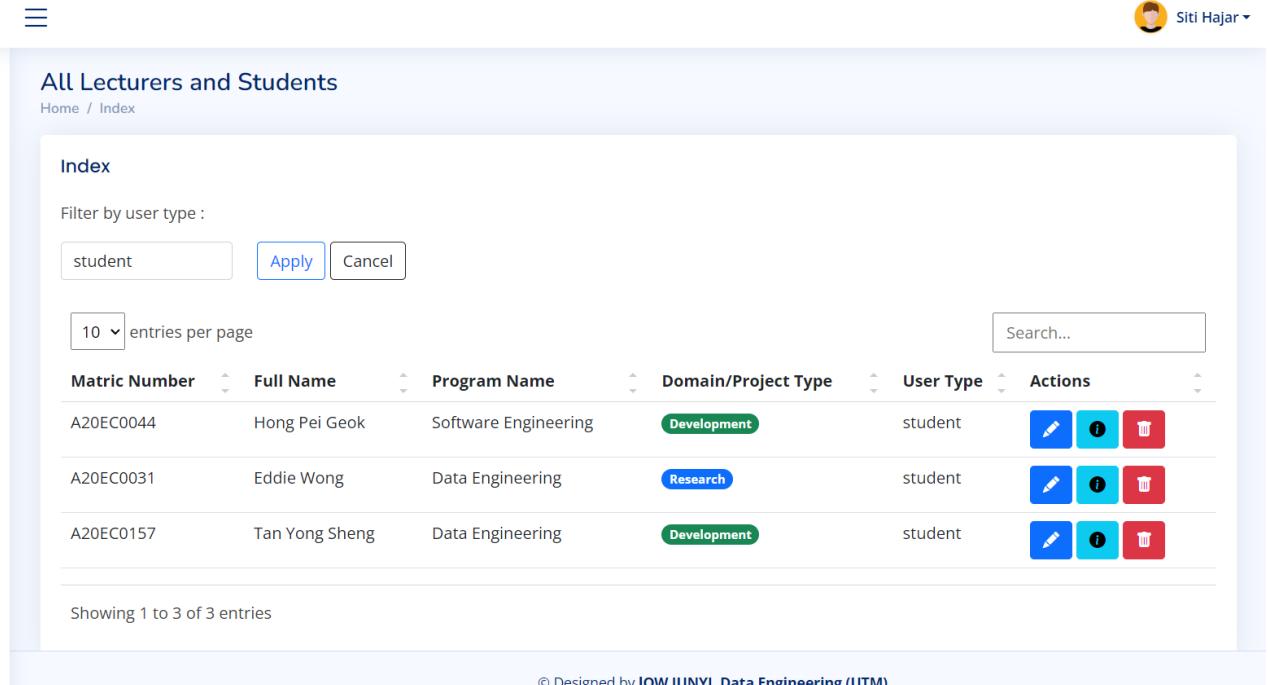


The screenshot shows the 'Read' interface for 'All Users'. The left sidebar has links for Home, Manage (All Users, PIC in Proposals, All Proposals), and a language switch (EN). The main content area is a table with columns: Matric Number, Full Name, Program Name, Domain/Project Type, User Type, and Actions. The table lists 8 entries. The 'Actions' column contains icons for edit, view, and delete. At the bottom, it says 'Showing 1 to 8 of 8 entries' and '© Designed by IOW JUNYI, Data Engineering (UTM)'.

Matric Number	Full Name	Program Name	Domain/Project Type	User Type	Actions
A20EC0044	Hong Pei Geok	Software Engineering	Development	student	
A20EC0031	Eddie Wong	Data Engineering	Research	student	
A20EC0157	Tan Yong Sheng	Data Engineering	Development	student	
12344	Alexander	Software Engineering	Development	lecturer	
12341	Siti Hajar	Data Engineering	Not yet assigned	committee	
34122	Iqbal Hakim	Data Engineering	Development	lecturer	
12122	Lim Mei Ying	Data Engineering	Development	lecturer	
23221	Farah Ann	Software Engineering	Research	lecturer	

Figure 4.31: Interface for <Read All Users In Systems>

Filter by user type: student



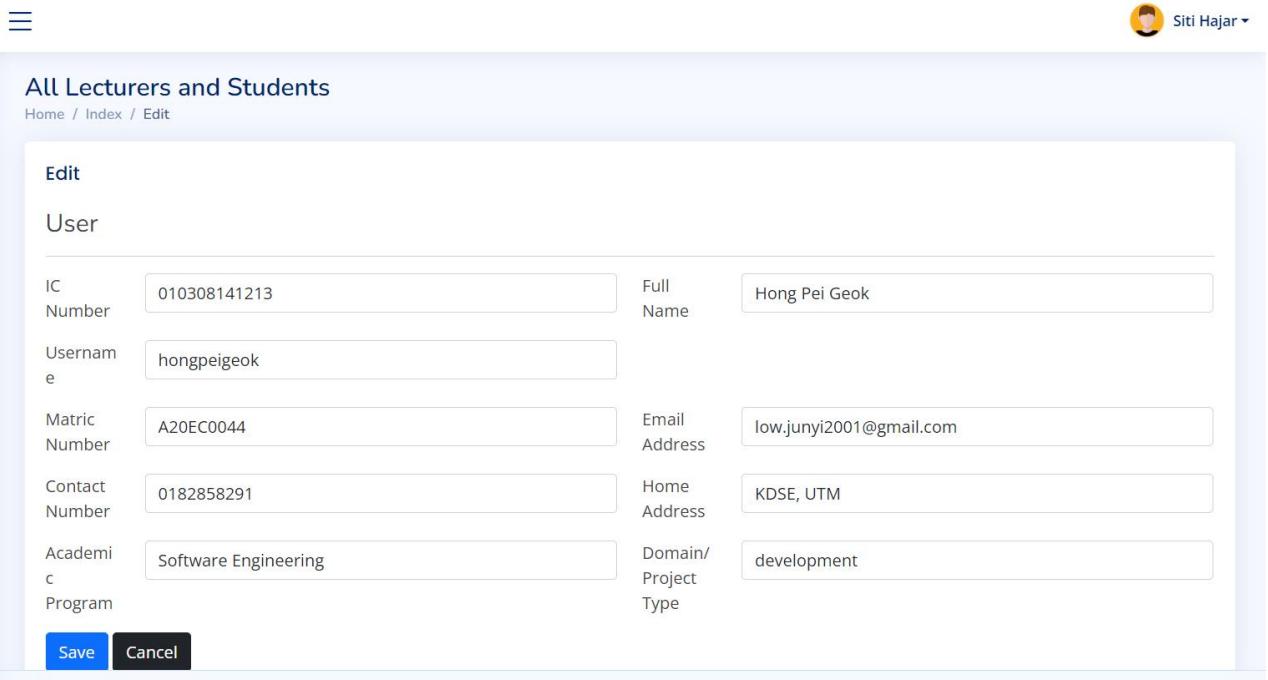
The screenshot shows a table of users with the following data:

Matric Number	Full Name	Program Name	Domain/Project Type	User Type	Actions
A20EC0044	Hong Pei Geok	Software Engineering	Development	student	  
A20EC0031	Eddie Wong	Data Engineering	Research	student	  
A20EC0157	Tan Yong Sheng	Data Engineering	Development	student	  

Showing 1 to 3 of 3 entries

Figure 4.32: Interface for <Read Users Filter by user type: student >

Update



The screenshot shows the 'Edit' form for a user with the following data:

IC Number	010308141213	Full Name	Hong Pei Geok
Username	hongpeigeok	Email Address	low.junyi2001@gmail.com
Matric Number	A20EC0044	Home Address	KDSE, UTM
Contact Number	0182858291	Domain/Project Type	development
Academic Program	Software Engineering		

Save Cancel

Figure 4.33: Interface for <Update Users>

Details

☰ Siti Hajar ▾

Home / Index / Details

Details	
IC Number	010308141213
Matric Number	A20EC0044
Full Name	Hong Pei Geok
Username	hongpeigeok
Contact Number	0182858291
Email Address	low.junyi2001@gmail.com
Home Address	KDSE, UTM
Program Code	SECJ
Domain/Project Type	Development
User Type	student

[Back](#)

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.34: Interface for <Read Details of Users>

Delete

☰ Siti Hajar ▾

Delete

Are you sure you want to delete this?

IC Number	010308141213
Matric Number	A20EC0044
Full Name	Hong Pei Geok
Username	hongpeigeok
Contact Number	0182858291
Email Address	low.junyi2001@gmail.com
Home Address	KDSE, UTM
Program Code	SECJ
Domain/Project Type	Development
User Type	student

[Delete](#) [Back](#)

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.35: Interface for <Delete Users>

Menu 2: PIC in Proposals

Read



Siti Hajar ▾

Index

Filter by Academic Session and Semester

10 ▾ entries per page
Search...

Academic Session	Semester	Student	Preferred Supervisor	Agreement	Evaluator 1	Evaluator 2	Actions
2022/2023	1	Hong Pei Geok	Lim Mei Ying Approved	limmei_yunzheng.pdf	Alexander	Iqbal Hakim	
2022/2023	1	Tan Yong Sheng	Alexander Pending	AD_IndividualTest_2023.pdf	Iqbal Hakim	Lim Mei Ying	
2022/2023	1	Eddie Wong	Farah Ann Pending	AD_IndividualTest_2023.pdf			

Showing 1 to 3 of 3 entries

© Designed by IOW JUNIYI, Data Engineering (UTM)

Figure 4.36: Interface for <Read PIC in Proposals>

Filter by academic 2022/2023 and semester :

Filter by Academic Session and Semester

2022/2023

1

10 ▾ entries per page

Search...

Academic Session	Semester	Student	Preferred Supervisor	Agreement	Evaluator 1	Evaluator 2	Actions
2022/2023	1	Hong Pei Geok	Lim Mei Ying Approved	limmei_yunzheng.pdf	Alexander	Iqbal Hakim	
2022/2023	1	Tan Yong Sheng	Alexander Pending	AD_IndividualTest_2023.pdf	Iqbal Hakim	Lim Mei Ying	
2022/2023	1	Eddie Wong	Farah Ann Pending	AD_IndividualTest_2023.pdf			

Figure 4.37: Interface for <Read PIC in Proposals Filter by academic 2022/2023 and semester>

Update



Siti Hajar ▾

Edit

Assign Evaluators

Student Name	Hong Pei Geok
Semester (eg. 1)	1
Academic Session (eg. 2022/2023)	2022/2023
Supervisor Name	Lim Mei Ying
Supervisor Approval Status	accepted
Evaluator 1 Name	Alexander
Evaluator 2 Name	Iqbal Hakim

Save **Back**

Figure 4.38: Interface for <Assign Evaluators>

Details



Siti Hajar ▾

Details

Academic Session	2022/2023
Semester	1
Student	Hong Pei Geok
Supervisor	Lim Mei Ying
Supervisor Agreement	limmei_yunzheng.pdf
Supervisor Approval Status	Approved
Evaluator 1	Alexander
Evaluator 2	Iqbal Hakim
Proposal Details	Proposal Details

Edit **Back**

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.39: Interface for <Read Details of all PIC in Proposals>

Delete



PIC in Proposal

Home / Index / Delete

Delete

Are you sure you want to delete this?

Academic Session (eg. 2022/2023)	2022/2023
Semester (eg. 1)	1
Student	Hong Pei Geok
Supervisor	Lim Mei Ying
Supervisor Agreement	limmei_yunzheng.pdf
Supervisor Approval Status	Approved
Evaluator 1	Alexander
Evaluator 2	Iqbal Hakim

Delete **Back**

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.40: Interface for <Delete all PIC in Proposals>

Menu 3: All Proposals

Read



Proposals

Home / Index

Index

Filter by Academic Session and Semester

Select Academic Sessions

10 entries per page

Student Proposal ID	Title	Proposal (only PDF is accepted)	Proposal Evaluation Result	
Student	Analytics on Data	YunZheng_A20EC0010_Proposal-1.pdf	Pending	
Student	Business Intelligence Success	sofia_A20EC0010_Proposal-3.pdf	Accepted	

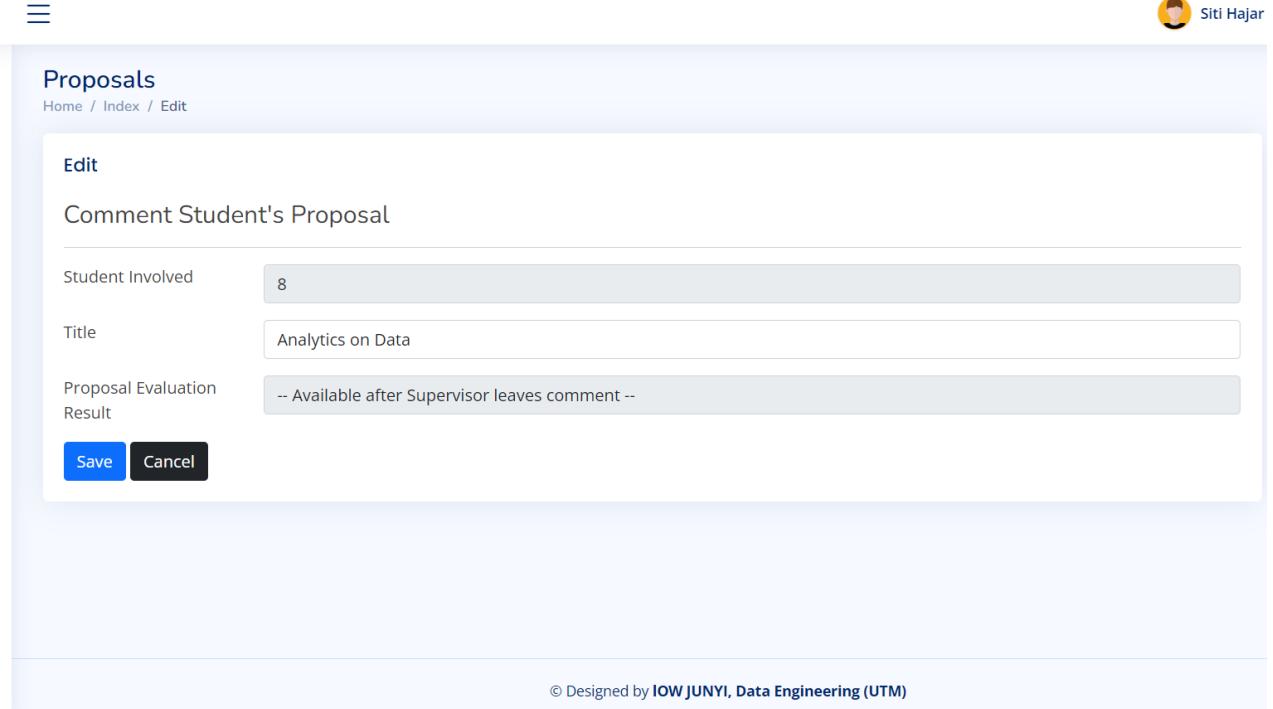
Showing 1 to 2 of 2 entries

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.41: Interface for <Read All Proposals>

Update

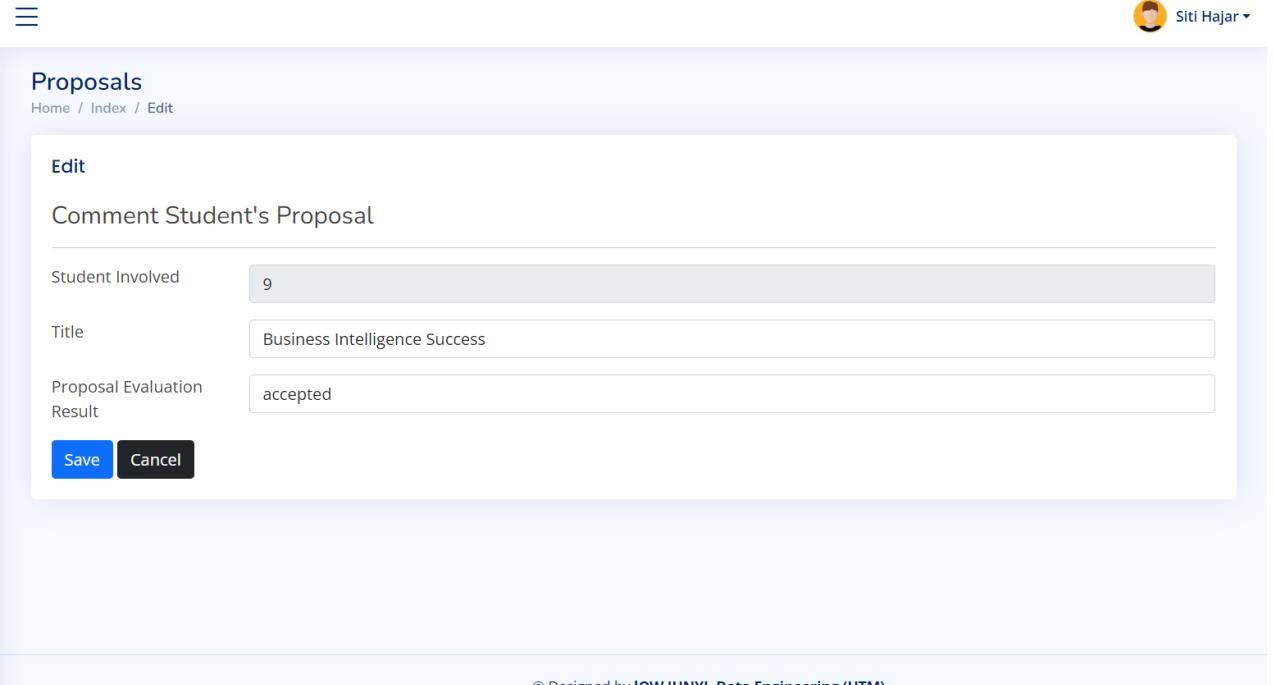
1. If the supervisor hasn't made a comment



The screenshot shows a 'Proposals' interface with a 'Comment Student's Proposal' form. The 'Student Involved' field contains the value '8'. The 'Title' field contains 'Analytics on Data'. The 'Proposal Evaluation Result' field contains the placeholder text '-- Available after Supervisor leaves comment --'. At the bottom are 'Save' and 'Cancel' buttons.

Figure 4.42: Interface for <Evaluate>

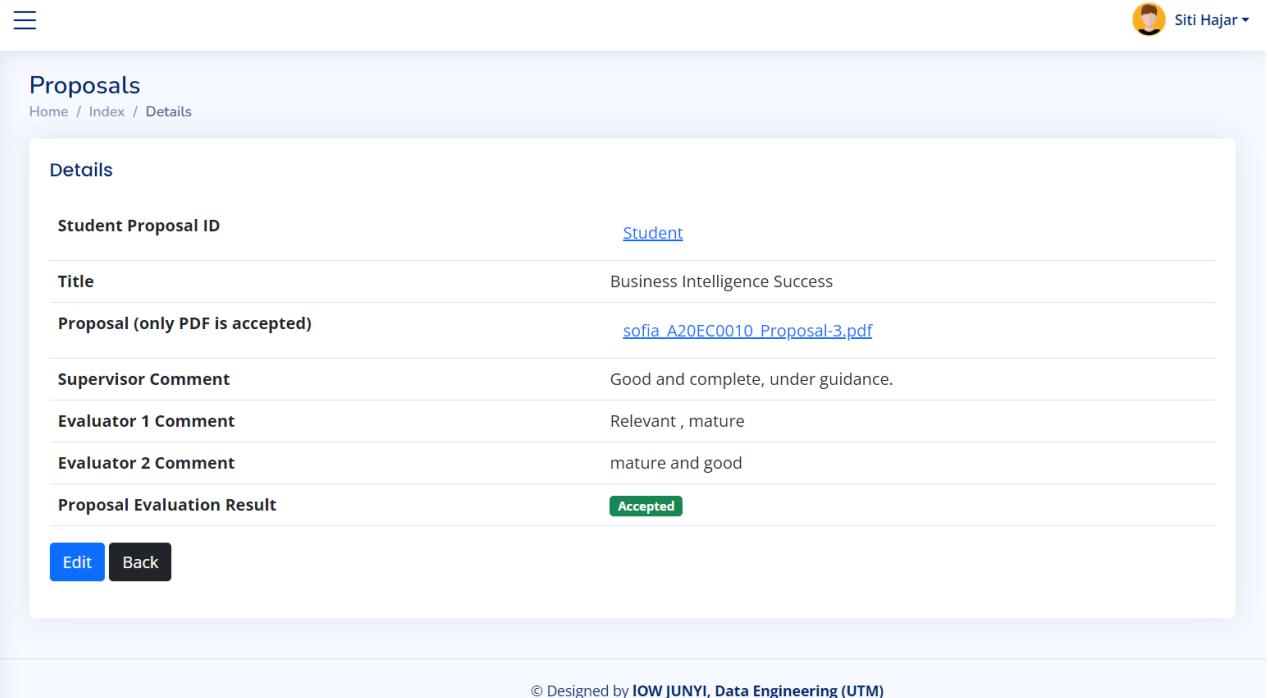
2. If a supervisor leaves comments



The screenshot shows a 'Proposals' interface with a 'Comment Student's Proposal' form. The 'Student Involved' field contains the value '9'. The 'Title' field contains 'Business Intelligence Success'. The 'Proposal Evaluation Result' field contains the value 'accepted'. At the bottom are 'Save' and 'Cancel' buttons.

Figure 4.43: Interface for <Evaluate>

Details



Proposals

Home / Index / Details

Details

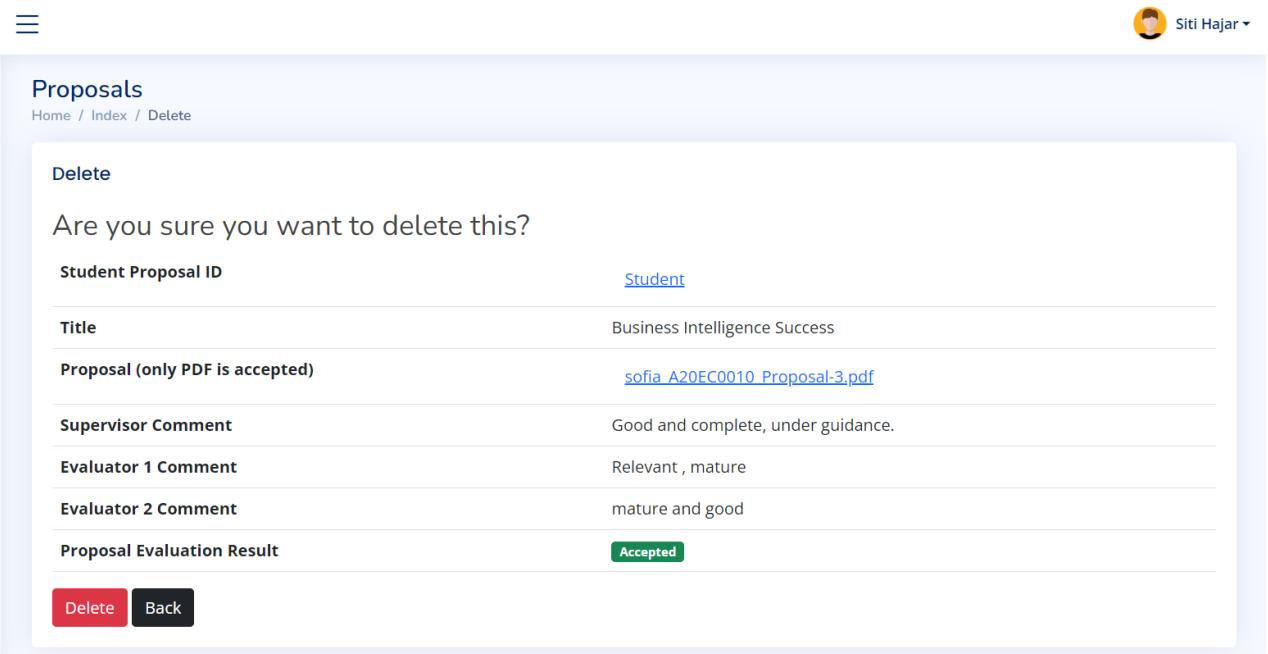
Student Proposal ID	Student
Title	Business Intelligence Success
Proposal (only PDF is accepted)	sofia_A20FC0010_Proposal-3.pdf
Supervisor Comment	Good and complete, under guidance.
Evaluator 1 Comment	Relevant, mature
Evaluator 2 Comment	mature and good
Proposal Evaluation Result	Accepted

Edit Back

© Designed by IOW JUNIYI, Data Engineering (UTM)

Figure 4.44: Interface for <View Proposal Details, Comments and Evaluation Result>

Delete



Proposals

Home / Index / Delete

Delete

Are you sure you want to delete this?

Student Proposal ID	Student
Title	Business Intelligence Success
Proposal (only PDF is accepted)	sofia_A20FC0010_Proposal-3.pdf
Supervisor Comment	Good and complete, under guidance.
Evaluator 1 Comment	Relevant, mature
Evaluator 2 Comment	mature and good
Proposal Evaluation Result	Accepted

Delete Back

© Designed by IOW JUNIYI, Data Engineering (UTM)

Figure 4.45: Interface for <Delete Proposal>

4.2.4 Lecturers (Supervisors/Evaluators)

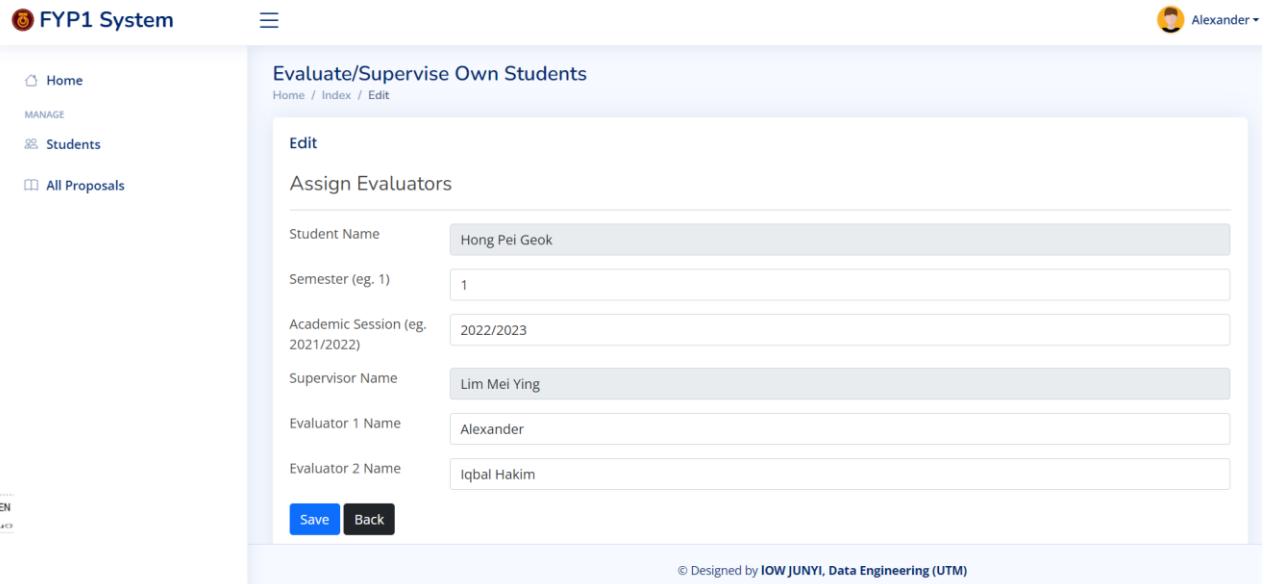
Home Page

Figure 4.46: Interface for <Lecturers (Supervisors/Evaluators)'s Home Page>

Menu 1: Students

Figure 4.47: Interface for <Read Own Students>

Update



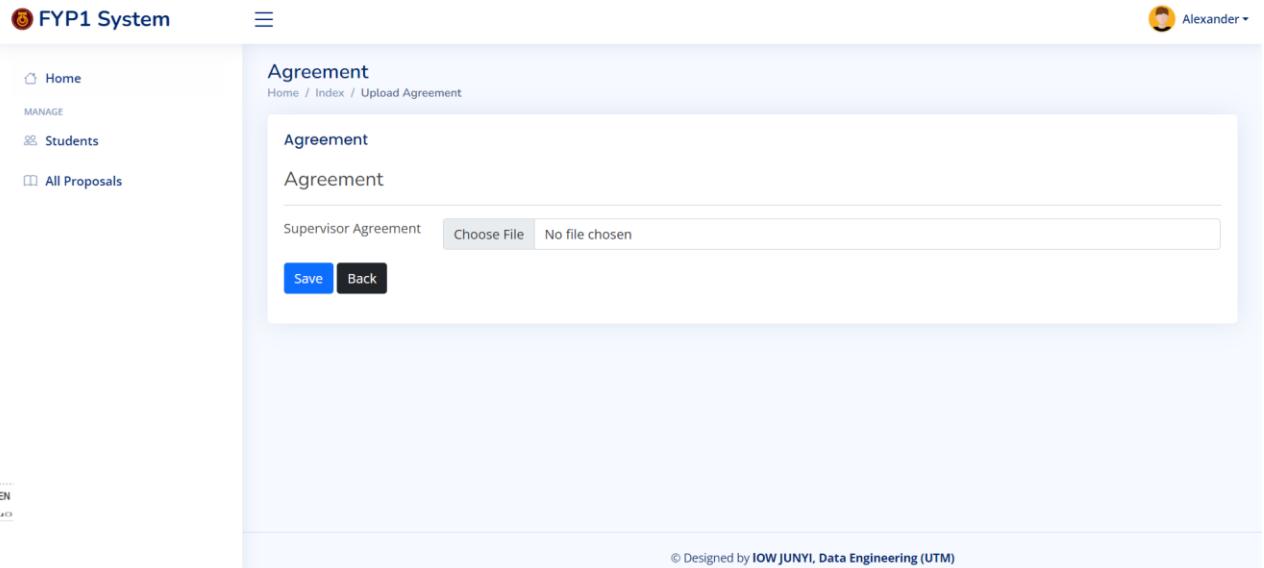
The screenshot shows a web-based application interface for managing student proposals. The top navigation bar includes a logo for 'FYP1 System', a user profile for 'Alexander', and a 'MANAGE' section with 'Home', 'Students', and 'All Proposals' links. A language switcher 'EN' is also present. The main content area is titled 'Evaluate/Supervise Own Students' and shows the 'Edit' form for 'Assign Evaluators'. The form fields are as follows:

Student Name	Hong Pei Geok
Semester (eg. 1)	1
Academic Session (eg. 2021/2022)	2022/2023
Supervisor Name	Lim Mei Ying
Evaluator 1 Name	Alexander
Evaluator 2 Name	Iqbal Hakim

At the bottom of the form are 'Save' and 'Back' buttons. A copyright notice at the bottom right reads '© Designed by IOW JUNYI, Data Engineering (UTM)'.

Figure 4.48: Interface for <Update Students PIC in Proposal>

Upload Agreement



The screenshot shows a web-based application interface for managing student proposals. The top navigation bar includes a logo for 'FYP1 System', a user profile for 'Alexander', and a 'MANAGE' section with 'Home', 'Students', and 'All Proposals' links. A language switcher 'EN' is also present. The main content area is titled 'Agreement' and shows the 'Upload Agreement' form. The form fields are as follows:

Supervisor Agreement	<input type="button" value="Choose File"/> No file chosen
----------------------	---

At the bottom of the form are 'Save' and 'Back' buttons. A copyright notice at the bottom right reads '© Designed by IOW JUNYI, Data Engineering (UTM)'.

Figure 4.49: Interface for <Upload Agreement>

Details

Evaluate/Supervise Own Students

Home / Index / Details

Details

Academic Session	2022/2023
Semester	1
Student	Hong Pei Geok
Supervisor	Lim Mei Ying
Supervisor Agreement	limmei_yunzheng.pdf
Supervisor Approval Status	Approved
Evaluator 1	Alexander
Evaluator 2	Iqbal Hakim
Proposal Details	Proposal Details

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.50: Interface for <Students Details>

Delete

PIC in Proposal

Home / Index / Delete

Delete

Are you sure you want to delete this?

Academic Session (eg. 2022/2023)	2022/2023
Semester (eg. 1)	1
Student	Hong Pei Geok
Supervisor	Lim Mei Ying
Supervisor Agreement	limmei_yunzheng.pdf
Supervisor Approval Status	Approved
Evaluator 1	Alexander
Evaluator 2	Iqbal Hakim

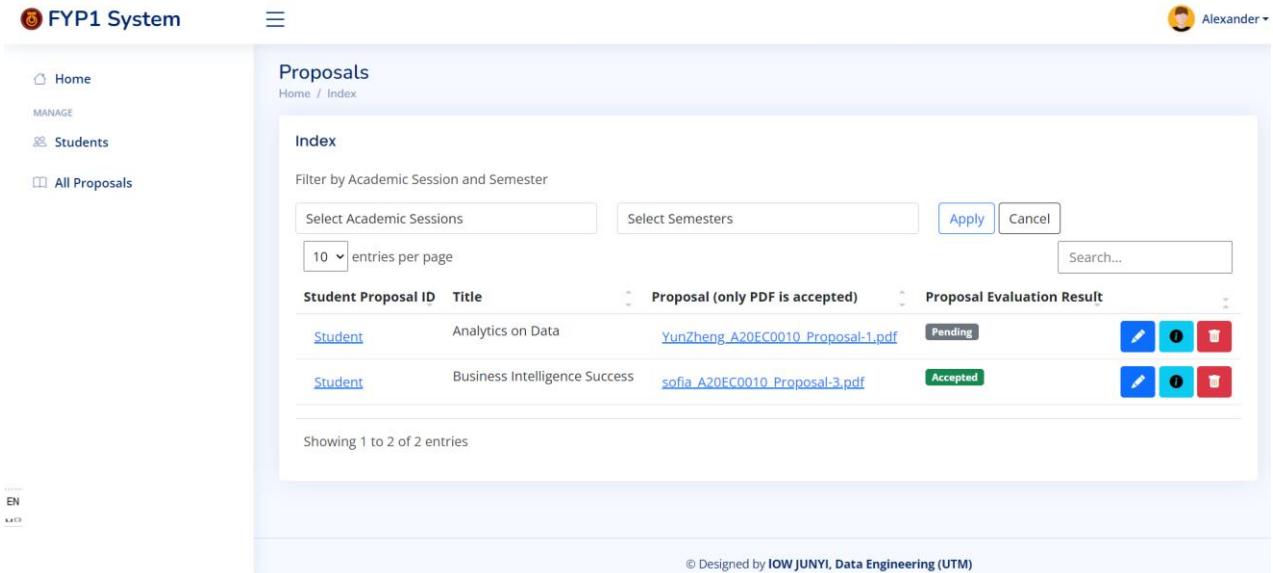
Delete **Back**

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.51: Interface for <Delete Student>

Menu 2: All Proposals

Read



The screenshot shows a web-based application interface for managing proposals. The top navigation bar includes a logo, the text 'FYP1 System', and a user profile for 'Alexander'. The left sidebar has links for 'Home', 'MANAGE', 'Students', and 'All Proposals'. The main content area is titled 'Proposals' and shows an 'Index' of entries. It includes filters for 'Select Academic Sessions' and 'Select Semesters', a dropdown for 'entries per page' (set to 10), and a search bar. The table displays two rows of data:

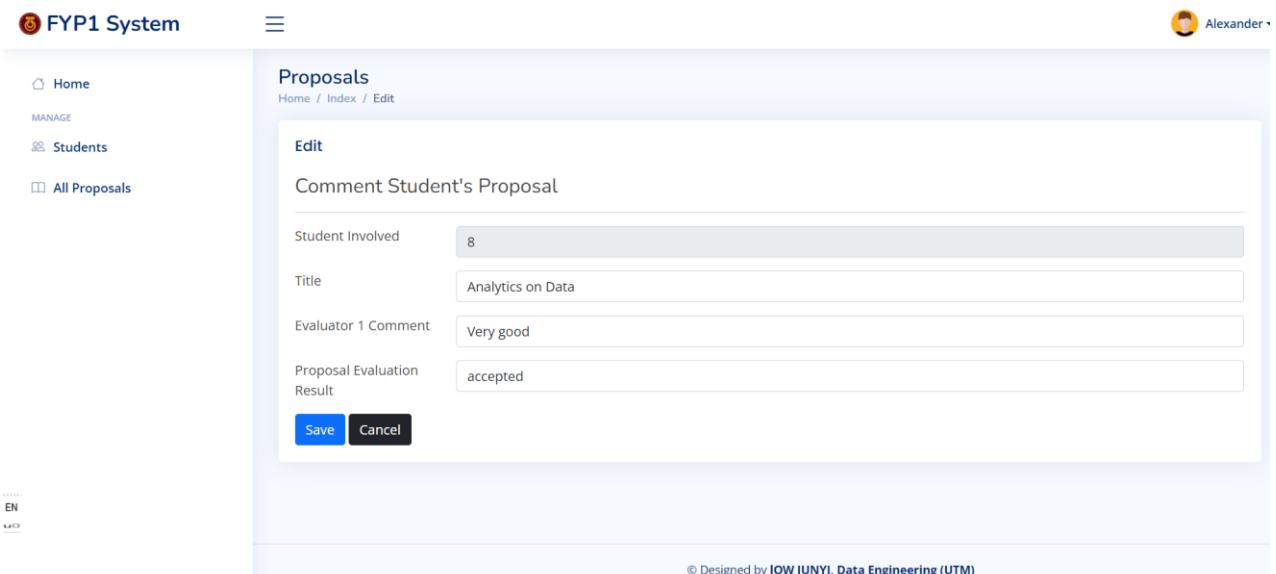
Student Proposal ID	Title	Proposal (only PDF is accepted)	Proposal Evaluation Result
Student	Analytics on Data	YunZheng_A20FC0010_Proposal-1.pdf	Pending
Student	Business Intelligence Success	sofia_A20FC0010_Proposal-3.pdf	Accepted

At the bottom, it says 'Showing 1 to 2 of 2 entries' and credits '© Designed by IOW JUNYI, Data Engineering (UTM)'.

Figure 4.52: Interface for <Read Own Students Proposals>

Update

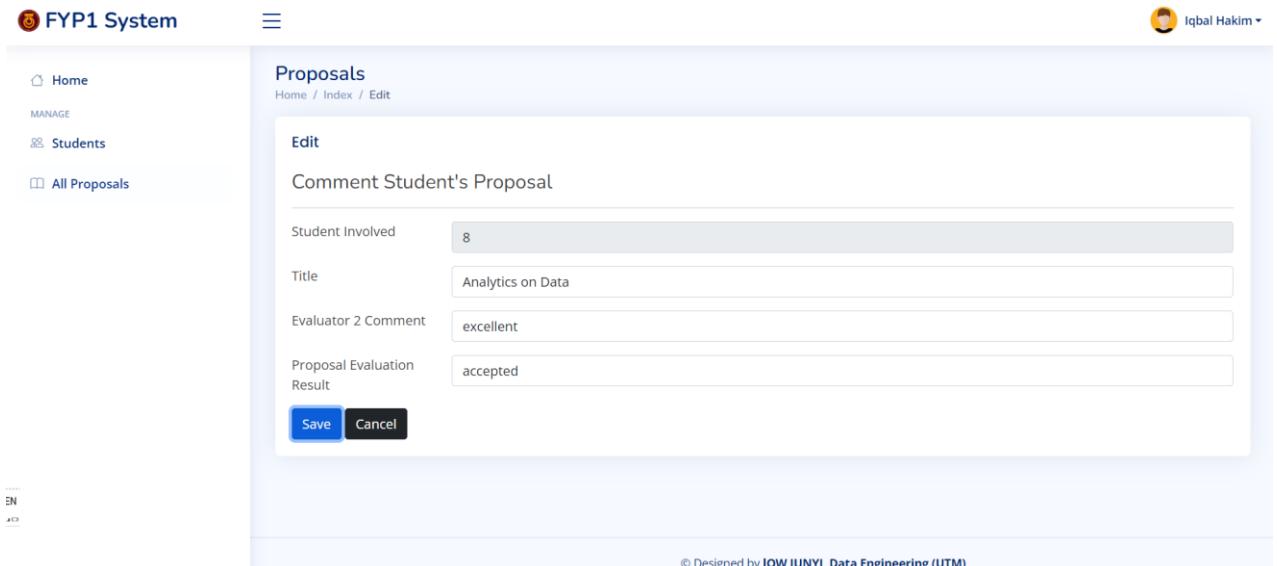
1. If the user is evaluator 1 and supervisor commented



The screenshot shows a 'Comment Student's Proposal' form. The top navigation bar includes a logo, the text 'FYP1 System', and a user profile for 'Alexander'. The left sidebar has links for 'Home', 'MANAGE', 'Students', and 'All Proposals'. The main content area is titled 'Edit' and shows the 'Comment Student's Proposal' form. It includes fields for 'Student Involved' (value: 8), 'Title' (value: Analytics on Data), 'Evaluator 1 Comment' (value: Very good), and 'Proposal Evaluation Result' (value: accepted). At the bottom are 'Save' and 'Cancel' buttons, and the credit '© Designed by IOW JUNYI, Data Engineering (UTM)'.

Figure 4.53: Interface for <Comment and Evaluate>

2. If the user is evaluator 2 and supervisor commented



Proposals
Home / Index / Edit

Edit

Comment Student's Proposal

Student Involved: 8

Title: Analytics on Data

Evaluator 2 Comment: excellent

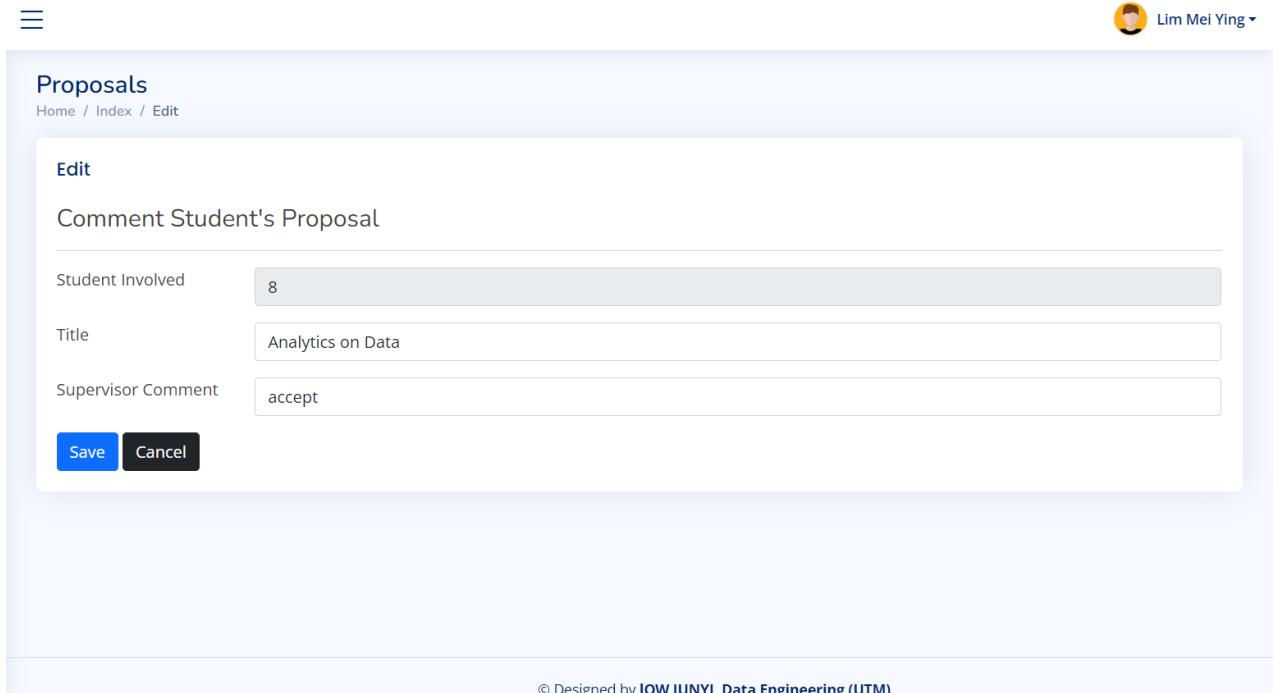
Proposal Evaluation Result: accepted

Save **Cancel**

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.54: Interface for <Comment and Evaluate>

3. If the user is a supervisor



Proposals
Home / Index / Edit

Edit

Comment Student's Proposal

Student Involved: 8

Title: Analytics on Data

Supervisor Comment: accept

Save **Cancel**

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.55: Interface for <Comment>

4. If the user is evaluator 1 and supervisor not commented

FYP1 System

Proposals

Comment Student's Proposal

Student Involved: B

Title: Analytics on Data

Evaluator 1 Comment: (empty)

Proposal Evaluation Result: -- Available after Supervisor leaves comment --

Save Cancel

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.56: Interface for <Comment and Evaluate>

5. If the user is evaluator 2 and supervisor not commented

FYP1 System

Proposals

Comment Student's Proposal

Student Involved: B

Title: Analytics on Data

Evaluator 2 Comment: (empty)

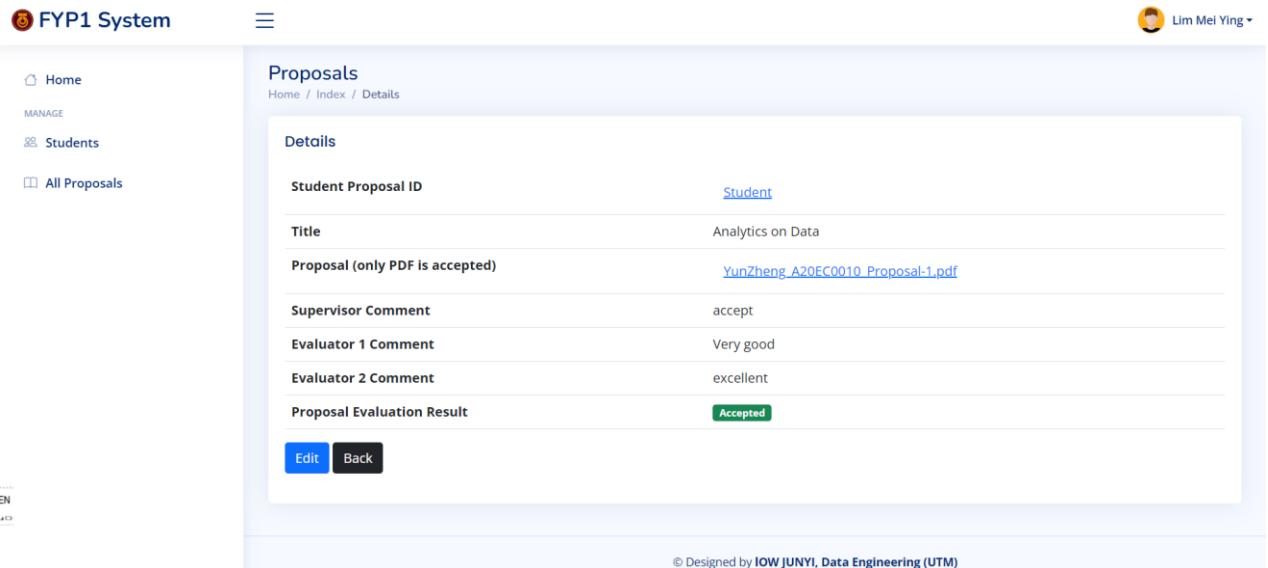
Proposal Evaluation Result: -- Available after Supervisor leaves comment --

Save Cancel

© Designed by IOW JUNYI, Data Engineering (UTM)

Figure 4.57: Interface for <Comment and Evaluate>

Details



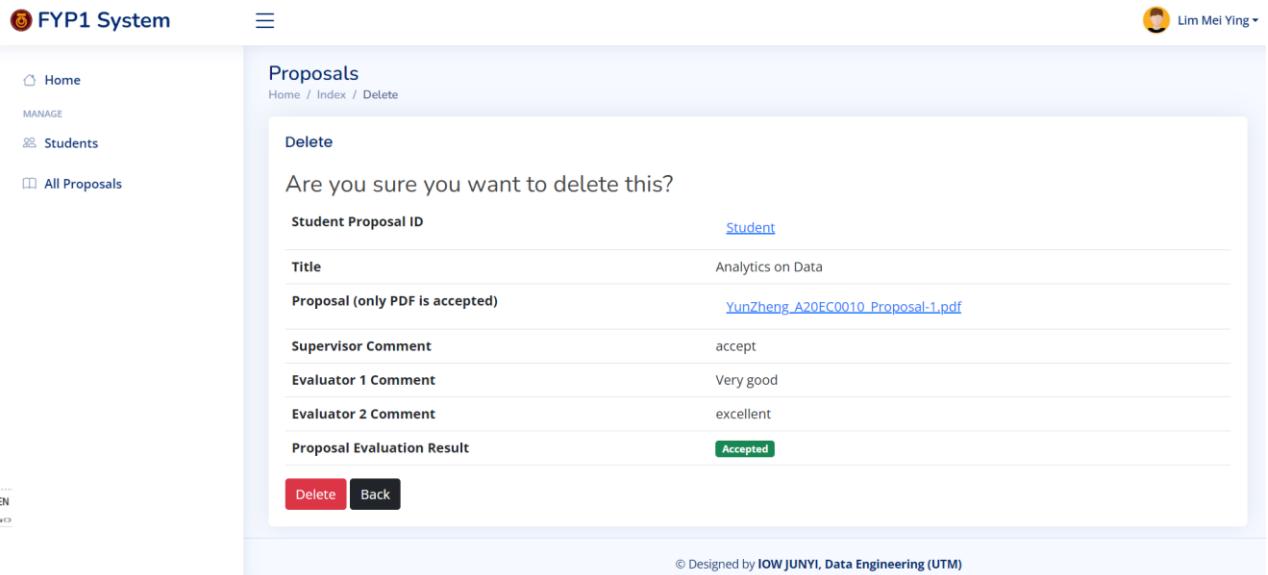
The screenshot shows the 'Details' page of the FYP1 System. The top navigation bar includes a logo, the text 'FYP1 System', and a user profile for 'Lim Mei Ying'. The left sidebar has links for 'Home', 'MANAGE', 'Students', and 'All Proposals'. The main content area is titled 'Proposals' and shows the 'Details' of a specific proposal. The proposal information includes:

Student Proposal ID	Student
Title	Analytics on Data
Proposal (only PDF is accepted)	YunZheng_A20EC0010_Proposal-1.pdf
Supervisor Comment	accept
Evaluator 1 Comment	Very good
Evaluator 2 Comment	excellent
Proposal Evaluation Result	Accepted

Buttons at the bottom are 'Edit' (blue) and 'Back' (black).

Figure 4.58: Interface for <View Proposal Details, Comments and Evaluation Result>

Delete



The screenshot shows a 'Delete' confirmation dialog within the FYP1 System. The top navigation bar and sidebar are identical to Figure 4.58. The main content area is titled 'Delete' and asks 'Are you sure you want to delete this?'. The proposal details are identical to Figure 4.58. Buttons at the bottom are 'Delete' (red) and 'Back' (black).

Figure 4.59: Interface for <Delete Proposal>

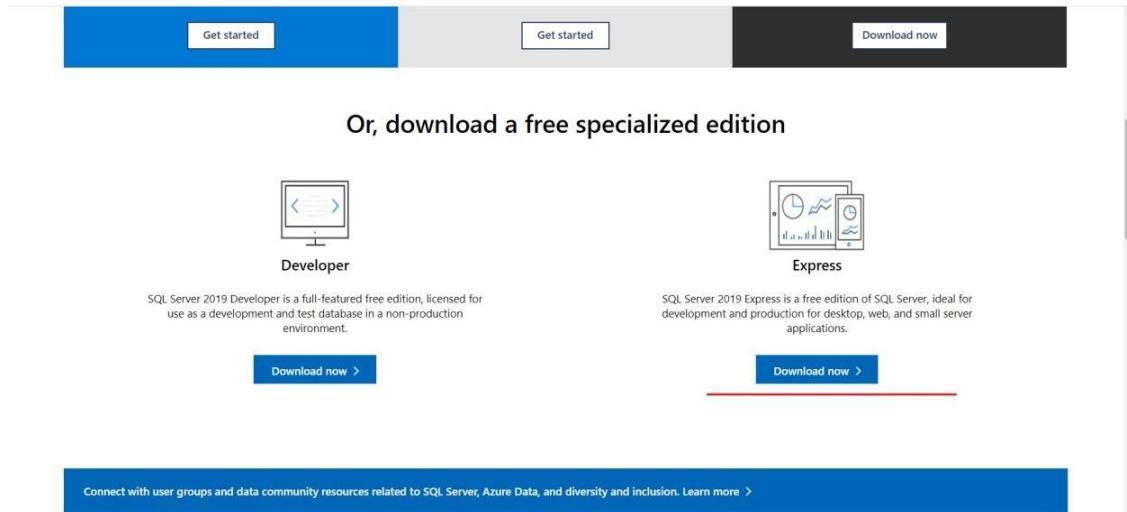
5.0 Localhost Installation Steps

Prerequisites:

1. Install Microsoft SQL Server Express Edition

Go to

<https://www.microsoft.com/en-us/sql-server/sql-server-downloads>

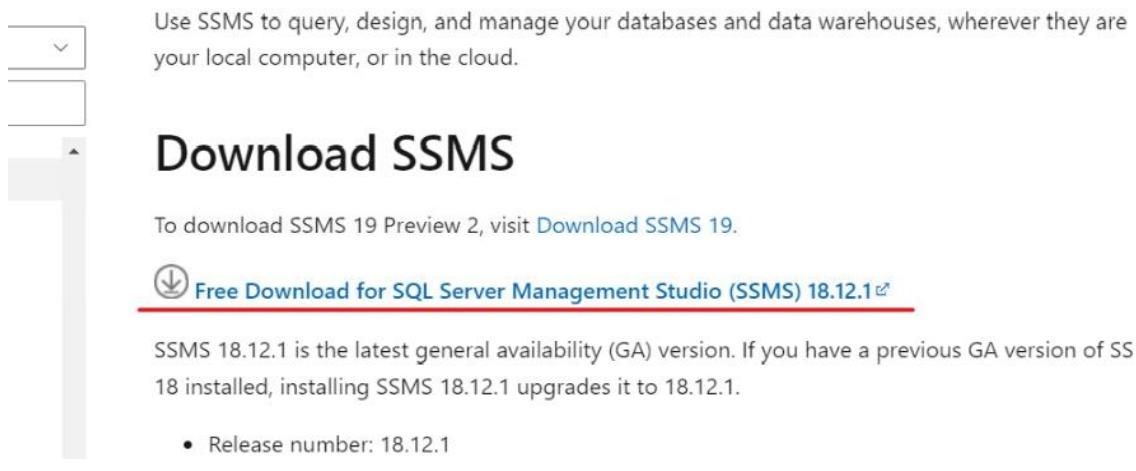


The screenshot shows the Microsoft SQL Server Downloads page. At the top, there are three main sections: 'Developer' (blue background, 'Get started' button), 'Express' (grey background, 'Get started' button), and 'Download now' (black background, 'Download now' button). Below these, there are two main download options: 'Developer' (with a monitor icon) and 'Express' (with a clock and chart icon). Each option has a 'Download now' button. A blue banner at the bottom of the page says 'Connect with user groups and data community resources related to SQL Server, Azure Data, and diversity and inclusion. Learn more >'.

Open the zipped file, click and select the Basic Version. Keep all paths default and install.

2. Install Microsoft SQL Server Management Studio (SSMS)

Go to <https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16>.

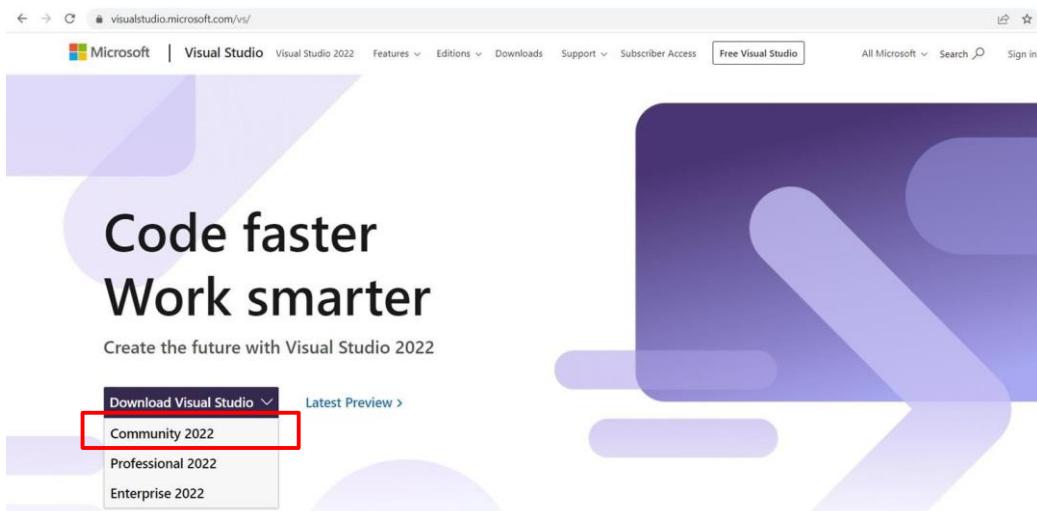


The screenshot shows the Microsoft SSMS download page. On the left, there's a sidebar with a dropdown menu. The main content area has a large 'Download SSMS' button. Below it, a red line highlights a 'Free Download for SQL Server Management Studio (SSMS) 18.12.1' link. A note says 'SSMS 18.12.1 is the latest general availability (GA) version. If you have a previous GA version of SS 18 installed, installing SSMS 18.12.1 upgrades it to 18.12.1.' A bullet point below says '• Release number: 18.12.1'.

Click the link above, run and install.

3. Install Microsoft Visual Studio 2022

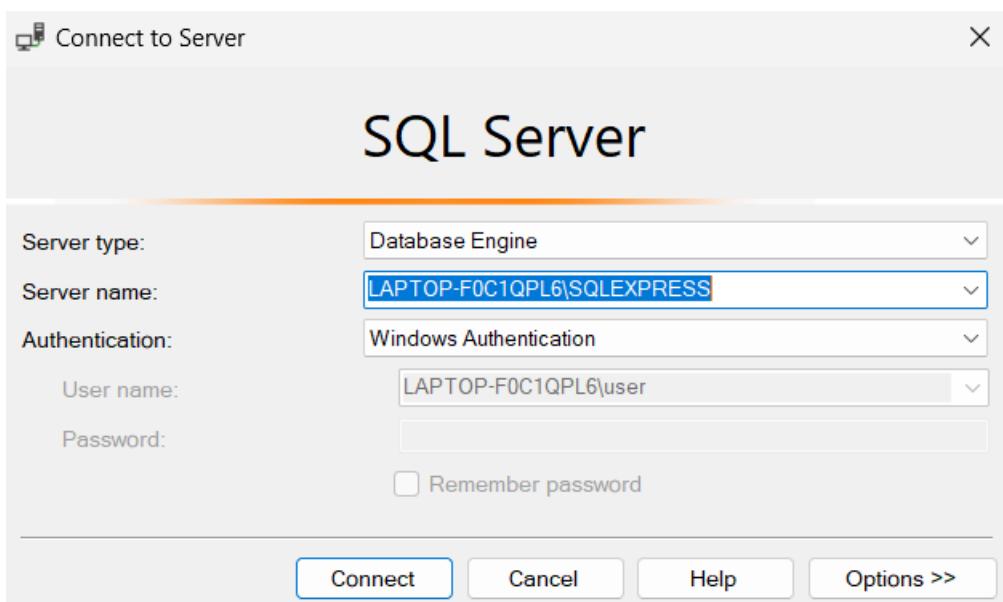
Go to <https://visualstudio.microsoft.com/vs/> and click the Community 2022 to download.



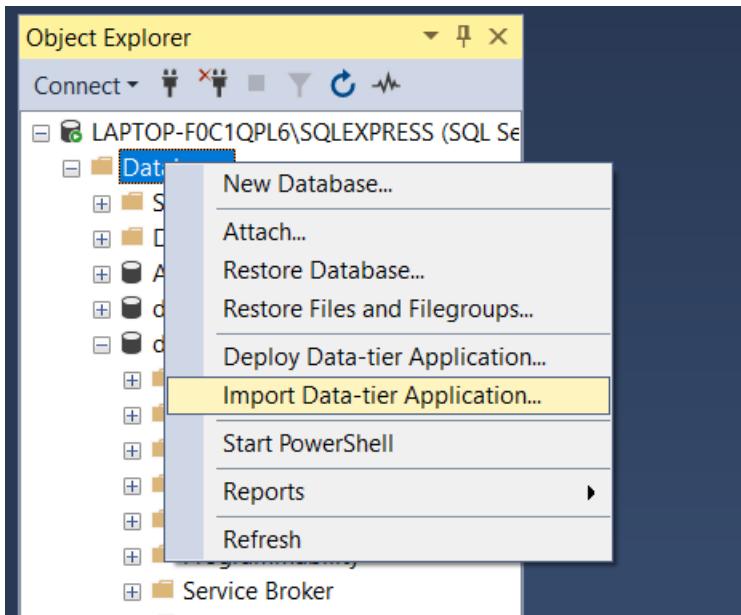
Run and install the applications.

Flows after manage to installation SSMS and Microsoft Visual Studio 2022:

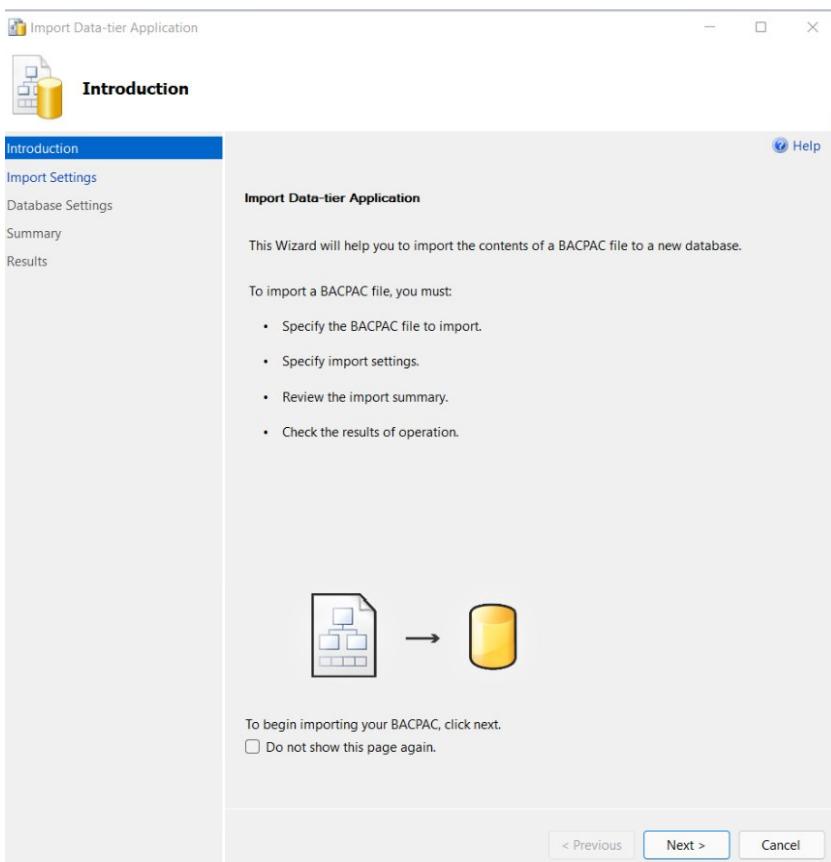
1. Extract the zipped file to your pc.
2. Open your Microsoft Server SQL Management Studio (SSMS).
3. Connect to SQL server by choosing your own server name. Copy the Server name as we have to use it later. Select “Window Authentication”. Click “Connect”.



4. Right-click “Databases” and click “Import Data-tier Application”.

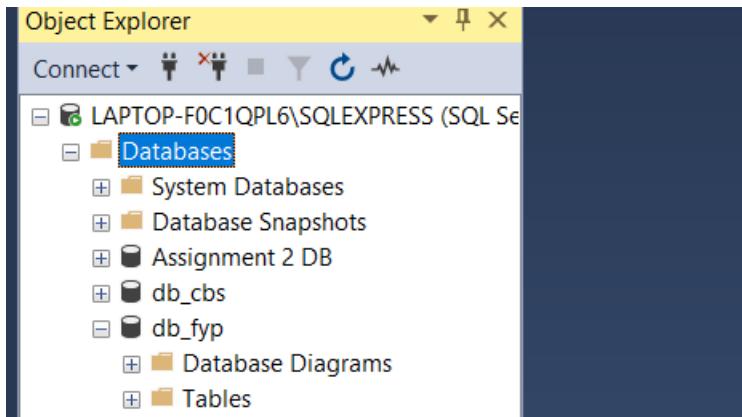


5. Click “Next”.

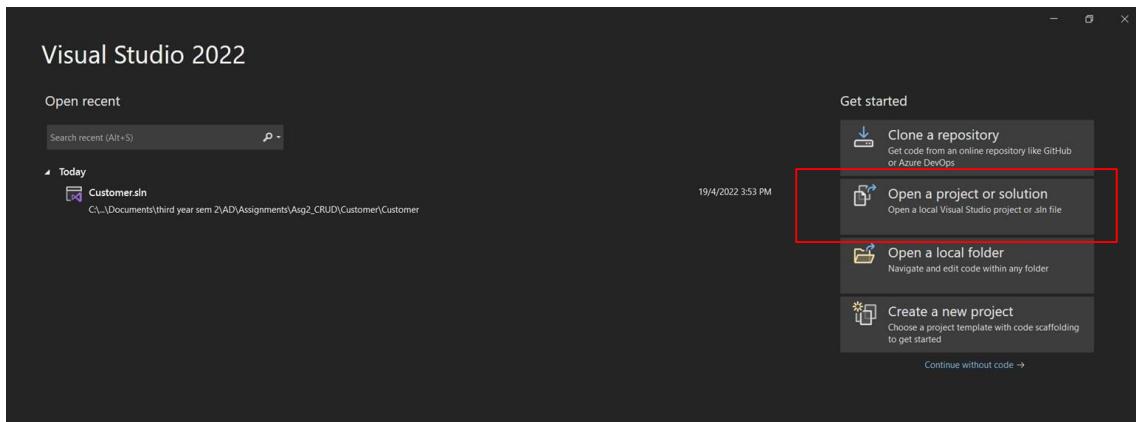


6. At the import from the local disk, click “Browse” and select the **db_fyp.bacpac** you extracted.

7. Click “Open”> “Next”. Ensure that the database name is “db_fyp”.
8. Click “Next” > “Finish”.



9. After import successfully, the db_fyp will appear under the folder Databases.
10. Then, Open Visual Studio 2022.
11. Select “Open project or solution”.

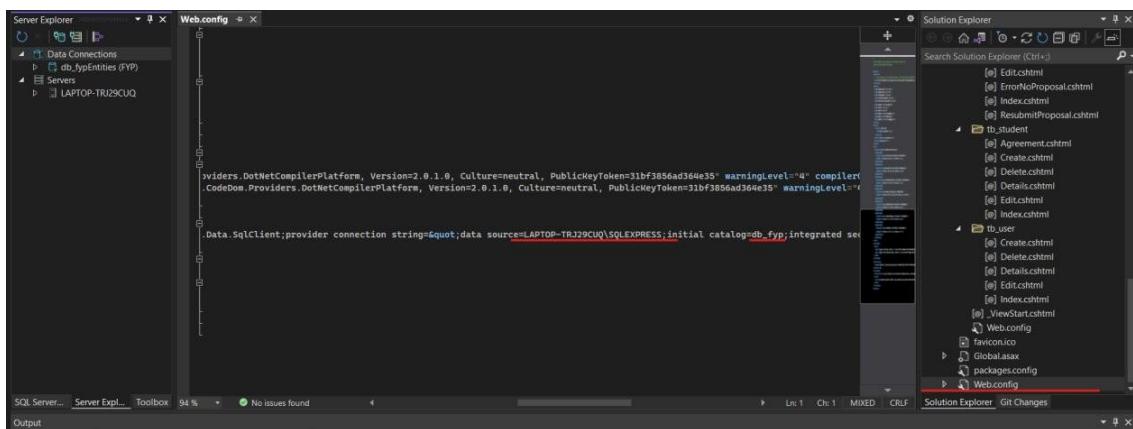


12. Browse to the file where you extract the FYP folder. Find **FYP.sln** inside the folder and open it.

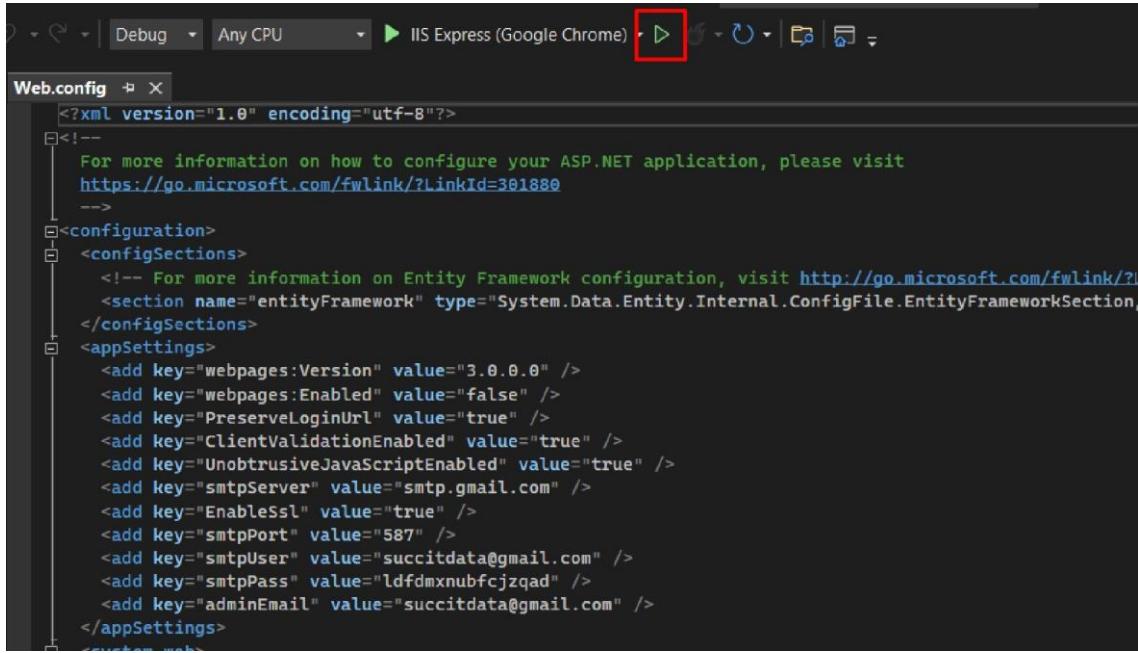
App_Data	21/6/2022 6:34 PM	File folder
App_Start	21/6/2022 6:34 PM	File folder
bin	29/6/2022 3:54 PM	File folder
Content	21/6/2022 6:44 PM	File folder
Controllers	1/7/2022 10:46 AM	File folder
Files	30/6/2022 11:43 PM	File folder
fonts	21/6/2022 6:34 PM	File folder
Models	1/7/2022 2:22 AM	File folder
obj	21/6/2022 6:34 PM	File folder
packages	22/6/2022 12:52 PM	File folder
Properties	21/6/2022 6:34 PM	File folder
Scripts	21/6/2022 6:34 PM	File folder
StudentProposals	1/7/2022 2:32 AM	File folder
SVagreement	1/7/2022 1:12 AM	File folder
Views	25/6/2022 3:42 PM	File folder
favicon	21/6/2022 6:34 PM	Icon
FYP	30/6/2022 10:41 PM	C# Project file
FYP.csproj.user	30/6/2022 10:41 PM	Per-User Project O...
FYP.sln	21/6/2022 6:34 PM	Visual Studio Solut...
Global.asax	21/6/2022 6:34 PM	ASP.NET Server Ap...
Global.asax.cs	22/6/2022 3:50 PM	C# Source File
packages	21/6/2022 8:58 PM	Configuration Sou...
Web	21/6/2022 10:13 PM	Configuration Sou...
Web.Debug	21/6/2022 6:34 PM	Configuration Sou...

13. Select Open.

14. Now, open the Web.config and change the data source at the bottom part according to the copied SQL server name. Make sure that the initial catalog value is the same with the imported database name.



15. Lastly, we can login to the system by clicking the triangle icon.

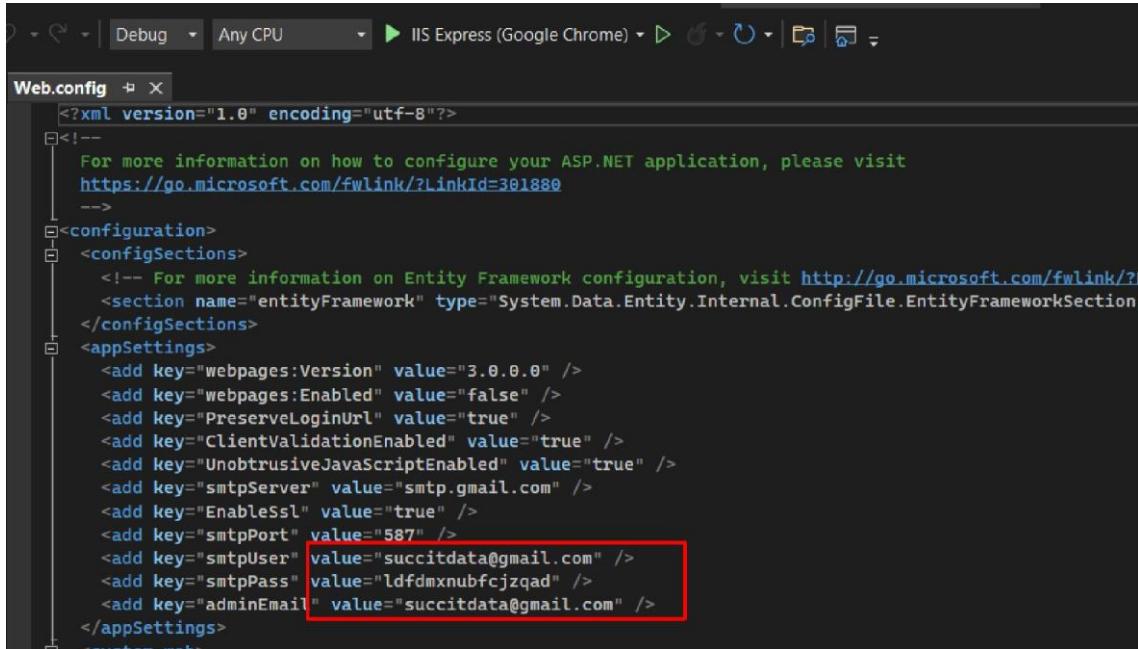


```
<?xml version="1.0" encoding="utf-8"?>
<!-- For more information on how to configure your ASP.NET application, please visit
     https://go.microsoft.com/fwlink/?LinkId=301880
-->
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/??
        <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection,
        </configSections>
  <appSettings>
    <add key="webpages:Version" value="3.0.0.0" />
    <add key="webpages:Enabled" value="false" />
    <add key="PreserveLoginUrl" value="true" />
    <add key="ClientValidationEnabled" value="true" />
    <add key="UnobtrusiveJavaScriptEnabled" value="true" />
    <add key="smtpServer" value="smtp.gmail.com" />
    <add key="EnableSsl" value="true" />
    <add key="smtpPort" value="587" />
    <add key="smtpUser" value="succitdata@gmail.com" />
    <add key="smtpPass" value="ldfdmxnubfcjzqad" />
    <add key="adminEmail" value="succitdata@gmail.com" />
  </appSettings>
</configuration>
```

Email sending feature (OPTIONAL)

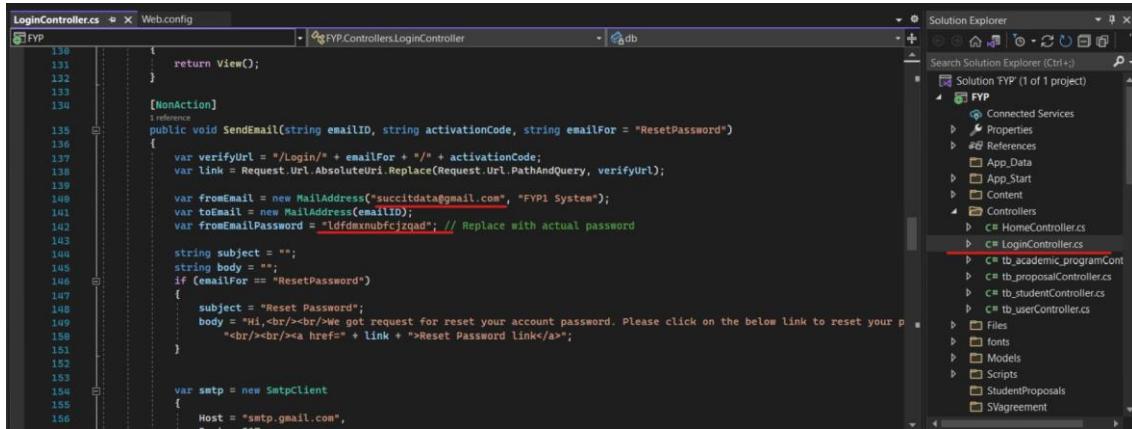
Please note that if this feature is not changed, the sender of email will be succIT when users click forgot password features.

1. In the same file (Web.config), change the last three values to your own email address and password.



```
<?xml version="1.0" encoding="utf-8"?>
<!-- For more information on how to configure your ASP.NET application, please visit
     https://go.microsoft.com/fwlink/?LinkId=301880
-->
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/??
        <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection,
        </configSections>
  <appSettings>
    <add key="webpages:Version" value="3.0.0.0" />
    <add key="webpages:Enabled" value="false" />
    <add key="PreserveLoginUrl" value="true" />
    <add key="ClientValidationEnabled" value="true" />
    <add key="UnobtrusiveJavaScriptEnabled" value="true" />
    <add key="smtpServer" value="smtp.gmail.com" />
    <add key="EnableSsl" value="true" />
    <add key="smtpPort" value="587" />
    <add key="smtpUser" value="succitdata@gmail.com" /> highlighted
    <add key="smtpPass" value="ldfdmxnubfcjzqad" /> highlighted
    <add key="adminEmail" value="succitdata@gmail.com" /> highlighted
  </appSettings>
</configuration>
```

2. Next, open the Controllers folder and double click LoginController.cs , at line 140 and 142, replace with your own email address and password.



```

130     {
131         return View();
132     }
133
134     [NonAction]
135     public void SendEmail(string emailID, string activationCode, string emailFor = "ResetPassword")
136     {
137         var verifyUrl = "/Login/*" + emailFor + "/" + activationCode;
138         var link = Request.Url.AbsoluteUri.Replace(Request.Url.PathAndQuery, verifyUrl);
139
140         var fromEmail = new MailAddress("sucildata@gmail.com", "FYP1 System");
141         var toEmail = new MailAddress(emailID);
142         var fromEmailPassword = "1dfdmxnbfczqad"; // Replace with actual password
143
144         string subject = "";
145         string body = "";
146         if (emailFor == "ResetPassword")
147         {
148             subject = "Reset Password";
149             body = "Hi,<br/><br/>We got request for reset your account password. Please click on the below link to reset your password<br/><br/><a href=" + link + ">Reset Password link</a>";
150         }
151
152
153
154         var smtp = new SmtpClient
155         {
156             Host = "smtp.gmail.com",
157             Port = 587,
158             EnableSsl = true,
159             DeliveryMethod = SmtpDeliveryMethod.Network,
160             Credentials = new NetworkCredential(fromEmail.Address, fromEmailPassword),
161             Timeout = 20000
162         };
163
164         MailMessage mail = new MailMessage(fromEmail, toEmail);
165         mail.Subject = subject;
166         mail.Body = body;
167
168         smtp.Send(mail);
169     }
170 }

```

3. Now, the sender of the email will be your email address with name “FYP1 System” when users click forgot password features. If your gmail is not allowed, try using yahoo or remain the default email address since gmail is restricting the usage of third party access.

6.0 User credentials

Username	Password	Role
lowjunyi	123456	Admin
peigeok	123456	Student (proposal uploaded)
tanyongsheng	123456	Student (proposal uploaded)
hazeek	123456	Student (supervisor selected, proposal not uploaded)
siti	123456	Committee
hakim	123456	Evaluator
alex	123456	Evaluator + supervisor to different students
limmei	123456	Evaluator + supervisor to different students

7.0 Documentation: SDD

Attached with this [link](#) in the same zipped file.