SW1 Projects Ideas

General Guidelines:

- Teams composed from 5-6 students who are registered for the course.
- Each Application should satisfy its software requirements
- All Applications must apply object oriented design. “O.O principles”
- Code must be well written and organized. (MVC Design Patterns)
- Database connection must apply Singleton Design Pattern
- Any Code written without understanding will allow student to get **Zero** in project discussion

All projects must have these operations:

- Add Operation
- Edit Operation
- Delete Operation
- Search Operation
- List Operation
- Authentication and Role Management

Each project must have these Documentation and Diagrams:

- Software Requirement Specification (SRS)
  - Functional Requirements
  - Non-Functional Requirements
- Use case Diagram
- Class Diagram
- Activity Diagram
- Sequence Diagram
- Database Tables (Database Schema)

Use **Git** version control in your project. Push your commits to Azure Repository or GitHub Repository.

Bonus:

You need to use Azure Devops to manage your project (Scrum Methodology)

- Setup a new team for your project. Invite a new member to work on project.
- Create your product backlog work items (Splitting your project into work items).
- Add these product backlogs to your Sprints (3 springs)

1- Vaccination scheduling System

Project description:

Vaccination in Egypt can be done by Ministry of health special vaccination units, or in some private entities. Vaccination Registration is tedious to manage manually, so there was a need for web application that can automate this whole process.

System must:

1. 3 User Roles: (User, Admin, Vaccination Center)
2. When opening the web application, you can know the available vaccination centers information. Each center has (Name, City, Address, Contact No., type(government or private))
3. Admin can:
   - Add City
   - Add Vaccination center: each center has (Name, City (dropdown menu), Address, Contact No. and type (government or private)). Also admin allot them username, passwords so that they can log in.
   - Add Vaccine: each vaccine has (Name, Gap between two doses (days), Precautions)
   - Search in Vaccination center by the city (to update or delete center)
   - List registered users on the system.
4. User can:
   - Register his information in the system (name, city, email, password, phone number, national id)
   - After login, user can see vaccination list.
   - Reserve vaccination: each reservation has (vaccination center, vaccine, first_dose_date or second_dose_date, user id)
   - User can’t reserve vaccine second dose until getting first dose
   - When reserving, the user will receive a vaccination reservation number

- After getting vaccine (2 doses), the user receive a certificate file that he has completed the vaccine
- When user reserve the second dose, the system should only allow the reservation if the date of the reservation is more than the gap between the 2 doses of the vaccine

5. Vaccination center can:
    - View user info by user vaccination reservation number and confirm that he has got the vaccine dose
    - List the users which has vaccination schedule this day.
    - After confirming the second dose of a user, a certificate file is uploaded to the user that has completed his vaccine

2- Freelance System

Project description:

A lot of work can be done remotely these days, so your dreams of working from your couch can come true by developing a freelance web application to manage all of these.

System must:

1. 3 User Roles: (Admin, Freelancer, Clients)
2. Wall page is like Home page in Facebook where posts is written it
3. Admin can:
   - Login to do his job
   - Add and Remove Users (Freelancers and Clients)
   - Each user (Freelancer or Client) has (first name, last name, phone number, profile image, user role). Also admin allot them username, passwords so that they can log in.
   - Accept or Refuse job posts written by the Clients before adding it to the Wall.
   - Remove and Update job posts in the wall page which are created by the client (after accepting them already)
   - If the Admin remove a post, it reflects directly to the wall and be removed.
4. Client can:
   - Login to do his job
   - See his profile information: (first name, last name, email, phone number, profile image)
   - Password Not Showed but we need a button to allow Client to Change his password

- Write job posts that will show in the wall page (through form). Admin Accept or Refuse Posts written by the Clients before adding it to the Wall.
- Each Post has (Client name, Job Type (fixed or hourly), Job Budget, post creation date, Job description, Number of proposal submitted).
- Show all his posts (History)
- Accept or Refuse freelancers’ proposals for a certain job. If the client accepts the proposal, the job post will be removed from the wall page.

5. Freelancer can:
    - View (read only) all job posts from different clients (no need to login)
    - Show all posts that have been added by Clients from their Profiles
    - Search for a job by Title, Date or Client name.
    - Save a particular post in saved page (to read later)
    - Apply to a job by sending a proposal to the client.

3- Faculty Management System

Project description:

Faculty Management System helps the faculty to communicate easily with students, to give students access to class documents, and for the convenience registering courses online.

System must:

1. 3 User Roles: (Admin, Professor, Student)
2. Admin can:
    - Add level (4 levels). Ex: lv1, lv2, lv3, lv4
    - Add subject: each subject has (name, code, description, level (dropdown menu), and semester).
    - Add Professor: each professor has (first name, last name, degree, gender, national id, date of birth, department). Also admin allot them username, passwords so that they can log in.
    - Create course: Allot subject to professor in available room and schedule this course in specific time at one day of the week. Course is 2 hours.
    - Search in professor list by the department (to update or delete center).
    - List registered student's data on the system.

3. Student can:
    - Register his information in the system (name, age, email, password, phone number, level)
    - After login, student can see courses list.
    - Enrolling course: student can register a course. Each enrollment has (course id, student id, others)
    - Student can't enroll in 2 courses which are scheduled in the same time

- Student can see his registered courses in time table (week time table)
- Upload answer file to the only question added by professor in specific course
- Student can see his transcript: student transcript is in a table form and has (registered courses and his grades)

4. Professor can:
    - View all his courses registered by the admin
    - Add only one question as a text for specific course and students upload there answers as files
    - List course students and add grade to student depends on his answer file

Best Wishes