

Task: You need to build a small web application that can add, view, update, and delete student details. The app should have a login system, use JWT for user session, and connect the frontend and backend using APIs.

Imp. Notes

Backend

- Use .NET (C#) for backend.
- Use MS SQL Server (SSMS) for database.
- Use JWT for login session.
- Use Swagger for automatic API documentation.
- All data should be handled through APIs only (no direct database calls from frontend).

Frontend

- Use simple HTML, CSS, vanilla JavaScript, AJAX, jQuery, and SweetAlert. You may use any library of your choice, such as Bootstrap, Tailwind, or Material UI.

Submission Details

- To Submit:
 - Create a new public Git repository named: visual_softech_your_name_frontend and visual_softech_your_name_backend.
 - Upload the respective code in each and share the links to both repositories.
 - In the backend repository, create a file named sql and include the full database creation script with all tables, views, and functions. Upload it as part of the backend code.

Advice: You must submit it within the deadline. Even if everything is not fully implemented, you should still submit it, as the objective is to evaluate your skill set.





Task Details

1. Login Page

- Show a simple login form with Username and Password.
- When login is successful, create a JWT token and allow the user to move to the main page.
- If login fails, show a Sweet Alert with a message like “Invalid Username or Password.”
- If the user is not logged in, they should not be able to open any page (index, create, or edit).

2. Index Page

- Show all students in a table with the following columns:
 - Name
 - Age
 - Address
 - State
 - Phone Number
 - Photo
 - Subjects
 - Actions (Edit / Delete)
- Delete Button: Show a Sweet Alert message: “Are you sure you want to delete this record?”. If user clicks “Yes,” delete data from both tables that is., student_master and student_detail
- Edit Button: Open the same form as the “Create Page” with all values already filled in.
- Implement pagination.

Student List							
Name	Age	Address	State	Phone Number	Photo	Subjects	Actions
Aarav Sharma	20	123 Park St	Mahara-shtra	+91-9876543210		Math, Physics	Edit
Ananya Iyer	21	456 Lake Rd	Karnataka	+91-8765432109		Biology, Chemistry	Edit
Rohan Verma	22	789 Hill View	Delhi	+91-7654321098		History, Geography	Edit
Priya Singh	19	101 Main St	Tamil Nadu	+91-6543210987		English, Economics	Edit

3. Create Page

- A form should include the following fields:
- Clicking Add will create a new row. Clicking Delete will remove that subject row.
- General student data will be saved in the student_mast table while Subject details will be saved in the student_detail table.
- User can upload multiple photos.
- Compress file to max 2kb before saving whenever user upload photos.
- To save data in database use sql adapter only.
- State Name will be a dynamic dropdown that is., data saved in state_name table values will come from.
- Add a "Save State Name" button, on clicking display modal where user can save state name if the list doesn't show his desired state. On clicking save only the state will be saved and not the student form.

Field	Type	Notes
Name	Text	Required
Age	Int	Required
Date of Birth	date	Required
Address	Text	Not Required
State	Int Dropdown	Required
Phone Number	Text	Must have regex validation
Photo(s)	File upload	Optional
Subjects	Table	User can add or delete rows dynamically

Create Student

Name*

Age*

Address*

State*

-- Select State --

Phone Number*

Photo

Choose File No file chosen

Subjects

Subject Name	Action
	<button>Add</button> <button>Delete</button>

Submit

4. Edit Page

- Same form as Create Page, but with prefilled data.
- When clicking Update, show a small modal that asks for a password. Password = 72991. If the password is correct update the record else show sweet alert: "Wrong Password."
- Data fetch api will return result in json array only.