

Student Registration Portal- Project Documentation

1. Objective

The **Student Registration Portal** is a web-based application designed to allow users to **register student details**, including **name, student ID, email, and contact number**. The system also provides functionalities for **editing and deleting records** while ensuring data persistence using **local storage**.

2. Functionalities Implemented

- Users can **add new student records** through a structured form.
- Existing student records can be **edited** with updated details.
- Students can be **deleted** from the list with a confirmation prompt.
- Data remains **persistent using local storage**, ensuring records do not disappear after refreshing the page.
- **Form validation** ensures:
 - Student ID and contact number accept only numbers.
 - Student name accepts only letters and spaces.
 - Email must follow a valid format.
- Users cannot submit the form with **empty fields**.
- A **vertical scrollbar** is dynamically added when student records exceed the visible specified size.

3. File Structure & Semantic Breakdown

File Structure

- `index.html` – Main HTML file
- `styles.css` – Styles for UI
- `script.js` – JavaScript logic
- `README.md` – Project documentation
- `Student-Registration-Portal.pdf`– Project documentation

HTML - (`index.html`)

- Used **semantic HTML elements** such as `header`, `section`, `table`, and `form` for accessibility.
- Contains a structured **student registration form**.
- Displayed student records in a **dynamic table format**.

- Used **meta tags** for SEO optimization and improved accessibility.

CSS - (**style.css**)

- Implemented a **responsive design** ensuring usability across **both mobile & desktop** devices.
- Used **CSS Grid & Flexbox** for structured alignment and layout management.
- **Scrolling enabled** for student records and table fields when exceeds size.
- **Sticky table headers** enhance visibility when scrolling.
- Ensures **consistent styling** with a light blue color palette and a clean layout.

JavaScript - (**script.js**)

- Implements **DOM manipulation** to dynamically update the student list.
- Handled **form validation** to prevent incorrect or incomplete submissions.
- Stored and retrieved data using **local storage** for persistence.
- Prevented **duplicate Student IDs** by checking before adding a new record.
- Options for **Adding, Update, Delete**.
- Ensures **students cannot be added with empty fields**.

4. Design Choices

- Designed with a **minimalist user interface** for simplicity and ease of use.
- Utilized a **consistent color scheme and structured form layout**.
- Provided **real-time feedback** through alerts for validation errors.
- Adapts to **different screen sizes** for accessibility on various devices.
- **Scrolling functionality** ensures ease of navigation for large datasets.
- **Sticky headers** maintain visibility of column titles when scrolling.

5. Challenges Faced & Solutions

Preventing Duplicate Student IDs

- Implemented a validation check to ensure unique student IDs.

Handling Form Reset After Editing

- Used an **editIndex** variable to track and manage edit operations and ui conditionally.

Maintaining Data After Page Refresh

- Integrated **local storage** to retain student records persistently.

Fixing UI Layout Issues on Mobile

- Applied **responsive media queries** to ensure the interface adapts effectively
- Enables horizontal and vertical scrolling for large data.

6. Live Demo & GitHub Repository

- Live Demo: <https://student-reg-portal.netlify.app/>
- GitHub Repository:
<https://github.com/SravanGunaganti/Student-Registration-Portal.git>

7. Conclusion

The **Student Registration Portal** is an efficient way to manage student records with **real-time updates and data persistence**. It is a beginner-friendly project for learning **JavaScript, DOM manipulation, and local storage**.