Student Name : Rajeshwaren AK

Register No. :710023104012

Subject : Web Technology

Academic Year / Semester :3rd Year

Date of Submission : 09-February 2026

College : Anna University Regional Campus Coimbatore,

Tamil Nadu

Aim

To design and develop a responsive web-based **Lost & Found Item Management Portal** that allows users to report lost or found items, browse listings, view item details, and facilitate communication between finders and owners — all using only **client-side technologies** (HTML, CSS, JavaScript) with localStorage for persistence.

Objectives

- Create a user-friendly interface for reporting lost and found items.
- Display separate listings for lost and found items with search-friendly layout.
- Allow users to view detailed information of any reported item.
- Simulate data persistence using browser localStorage so newly reported items appear in listings.
- Demonstrate responsive design using pure CSS (Grid + Flexbox).
- Provide clear navigation and professional look suitable for college campus use.
- Document the complete development process for academic submission.

Software Requirements

Development Tools / Environment

- Operating System : Windows 11 / macOS / Linux
- Code Editor : Visual Studio Code
- Browser : Google Chrome / Microsoft Edge (for testing & debugging)
- Version Control : Git + GitHub (for version tracking & hosting)

Technologies Used

- **Frontend** : HTML5, CSS3, JavaScript (Vanilla)
- **Styling** : Pure CSS (Flexbox + CSS Grid, no Bootstrap/Tailwind)
- **Data Storage** : JavaScript array + browser localStorage (client-side persistence)
- **Hosting (optional)** : GitHub Pages / Netlify (static deployment)
- **Libraries** : None (100% vanilla implementation)

Hardware Requirements: Any standard laptop/desktop with 4GB+ RAM and modern browser

System Architecture / Website Structure

Folder Structure

```
lost-found-portal/
├── index.html           → Home page with recent previews
├── lost.html            → Lost items listing
├── found.html           → Found items listing
├── report.html          → Form to report new lost/found item
├── details.html         → Single item detail view
├── about.html           → Project documentation & credits
│
├── css/
│   └── style.css        → All custom styles (pure CSS)
│
├── js/
│   ├── data.js          → Sample static items array + helpers
│   └── script.js        → Shared logic (filtering, merging localStorage,
utilities)
│
├── images/
│   └── items/           → Placeholder images for sample items
│
└── README.md
```

High-Level Architecture

```
User → Browser (HTML + CSS + JS)
       ↓
     Static Files (GitHub Pages / Netlify)
       ↓
     JavaScript
       ├── Static sample data (data.js)
       └── Dynamic user reports (localStorage)
       ↓
     Render dynamic listings & details
```

Procedure / Methodology

1. **Planning**
   - Identified real-world problem: lost items in college campus
   - Listed required pages and features
   - Decided to keep it static + localStorage for academic demo
2. **Design**
   - Created responsive layout using CSS Grid for cards
   - Designed consistent navbar with checkbox hamburger for mobile
   - Planned card-based UI for items (image + title + location + date)
3. **Development**
   - Created HTML structure for all pages
   - Wrote pure CSS (no framework) for styling and responsiveness
   - Implemented static sample data in data.js
   - Added form handling in report.html with localStorage save
   - Created merge logic in script.js so user reports appear in listings
   - Made details.html read item by ID from combined data
4. **Testing**
   - Tested in Chrome, Edge, Firefox
   - Checked mobile responsiveness (hamburger menu, grid layout)
   - Verified that new reports appear in correct tab (lost/found)
   - Cleared localStorage to simulate fresh start
5. **Deployment**
   - Pushed to GitHub repository
   - Enabled GitHub Pages for live demo
6. **8. Modules / Pages Description**

Page	Purpose	Key Features
index.html	Home / Landing page	Intro + Recent 3 lost + 3 found previews
lost.html	Displays all lost items	Card grid, “Report Lost” button
found.html	Displays all found items	Card grid, “Report Found” button
report.html	Form to report new lost or found item	Radio (lost/found), validation, localStorage save
details.html	Shows full details of one item	Image, description, contact, back button
about.html	Project documentation & credits	Aim, technologies, advantages, developer info

Code Snippets

A. Sample data (data.js)

JavaScript

```
const items = [
  { id: 1, type: "lost", title: "Black Earbuds", ... },
```

```
  // more sample items
];
```

B. Merge logic – script.js (key part)

JavaScript

```
function getAllItemsIncludingUserSubmitted() {
  let all = [...items];
  const stored = localStorage.getItem('lostFoundItems');
  if (stored) all = [...all, ...JSON.parse(stored)];
  all.sort((a,b) => new Date(b.date) - new Date(a.date));
  return all;
}

function getItemsByType(type) {
  return getAllItemsIncludingUserSubmitted().filter(i => i.type === type);
}
```

C. Report form save (report.html)

JavaScript

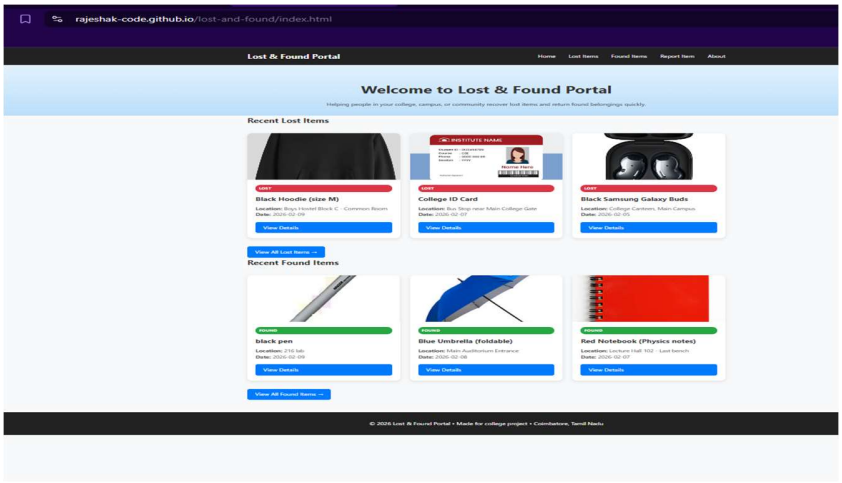
```
const newItem = { id: 'user-'+Date.now(), type: formData.get('type'), ... };
let reports = JSON.parse(localStorage.getItem('lostFoundItems') || '[]');
reports.push(newItem);
localStorage.setItem('lostFoundItems', JSON.stringify(reports));
```

D. Card rendering example (lost.html / found.html)

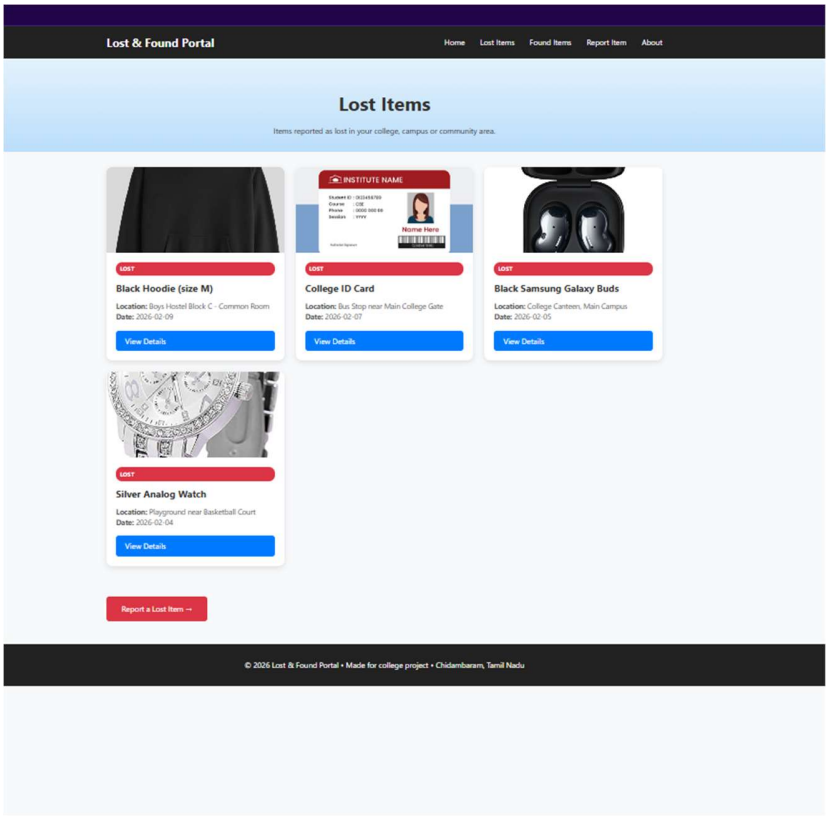
JavaScript

```
grid.innerHTML = items.map(item => `
  <div class="card">
    <img src="${item.image || 'placeholder'}">
    <span class="badge ${item.type === 'lost' ? 'badge-lost' : 'badge-
found'}">${item.type.toUpperCase()}</span>
    <h3>${item.title}</h3>
    ...
  </div>
`).join('');
```

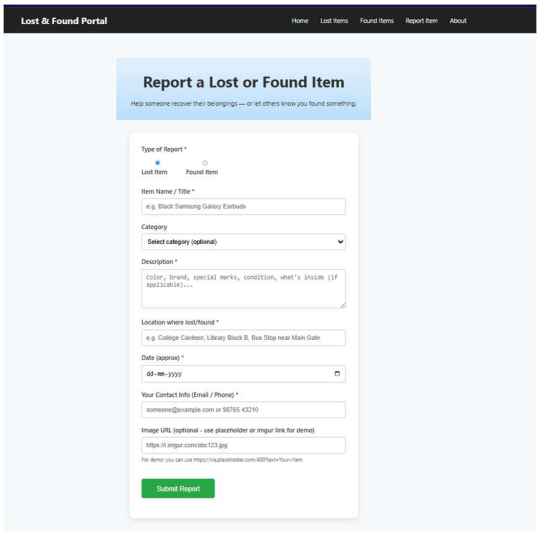
10. Output Screenshots



Home



Lost



Report

Future Scope

- Integrate real backend database (Supabase / Firebase / MongoDB)
- Add user authentication (login/signup)
- Support actual image upload (to Supabase Storage / Cloudinary)
- Implement real-time updates (new reports appear without refresh)
- Add search bar, category filter, date range filter
- Email notification to contact person when item is reported
- Admin panel to mark items as “Recovered” or delete spam
- Mobile app version using same backend

Conclusion

- The **Lost & Found Portal** successfully demonstrates how modern web technologies (HTML5, CSS3 Grid/Flexbox, JavaScript, localStorage) can be used to build a practical, responsive, and user-friendly application without any backend server.
- The project achieves its aim of providing a centralized platform for reporting and viewing lost & found items in a college environment. By combining static sample data with dynamic user-submitted reports (via localStorage), it gives a realistic demo experience while remaining lightweight and free to host.
- This project helped in understanding client-side development, responsive design principles, DOM manipulation, form handling, and data persistence in the browser — key skills in Web Technology.