**Ship Maintenance Dashboard**

**Objective:**

A front-end-only React application, the Ship Maintenance Dashboard replicates an entire ship maintenance system. Role-based access (Admin, Inspector, Engineer) and user authentication are included, and users can oversee ships, their parts, and maintenance tasks. In-app notifications are triggered by actions, and jobs are shown in a calendar view. Key statistics are displayed on a KPI dashboard, and all data is kept in localStorage. The application works flawlessly on all devices.

**Core Features:**

Here are the core features of the Ship Maintenance Dashboard

* used localStorage-based session handling to simulate authentication.
* Organize ships by adding, editing, and deleting them.
* Assign and monitor the parts associated with every ship.
* Create maintenance tasks and keep track of them with status updates.
* Use the calendar view to see the planned maintenance.
* alerts for impending and past-due tasks.
* KPI dashboard displaying important data, such as job completion rates.
* localStorage in the browser for persistent data storage.

Tech Stack Tool Objective:

1. React Router Page routing/navigation
2. Custom CSS localStorage
3. React UI Framework

**Installation:**

Setting up

1.1. Download the Repository

📁 2. Navigate into the Project Directory

    cd ship-maintenance-dashboard

🛠️ 3. Run Terminal as Administrator(optional)

* Right-click on Command Prompt or PowerShell
* Click “Run as Administrator” (important if script policies block npm)

📦 4. Install Required Packages

    npm install --force

    npm install --force

is used to bypass certain warnings or dependency conflicts. It is optional and should be used with care.

🔍 5. Fix Vulnerabilities (Optional)

    npm audit fix

▶️ 6. Start the Development Server

    npm start

    The app will launch at http://localhost:3000

**Usage for this project:**

You must log in using specific credentials for three different roles on the homepage.

For the admin:

     email address: admin@entnt.com;

     password: admin123

For the Engineer

     email address : engineer@entnt.com,

     password : engine123.

For the inspector

     Email:inspector@entnt.com,

     password is inspect123.

 Depending on their roles, everyone has restricted access.

 For example, an administrator can create jobs, update jobs, remove jobs, and create ships, as well as perform all operations on the site;

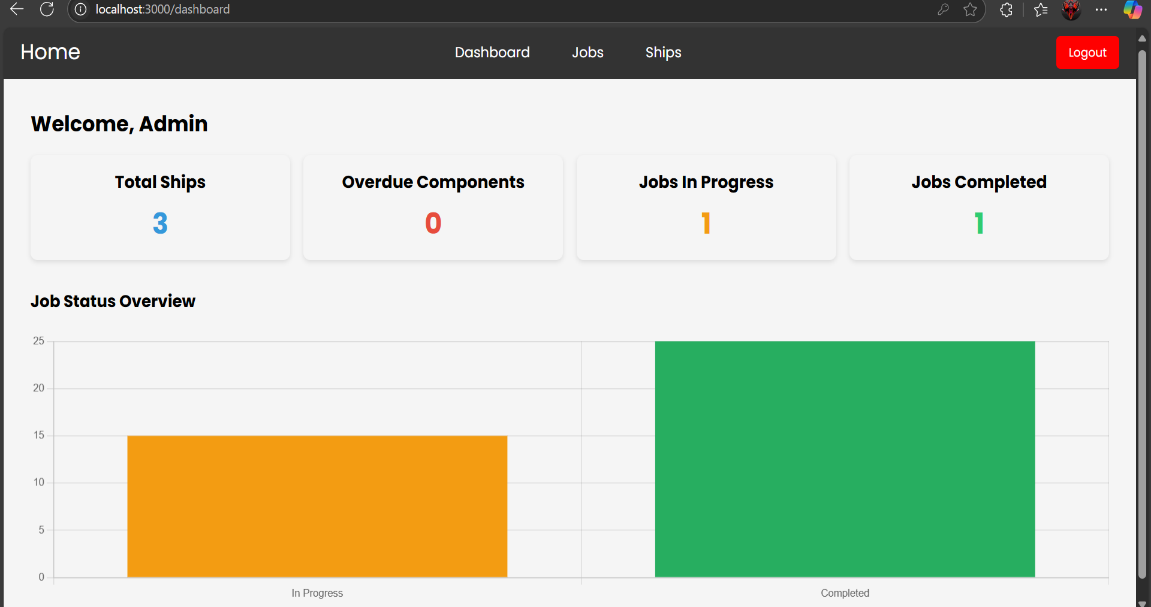
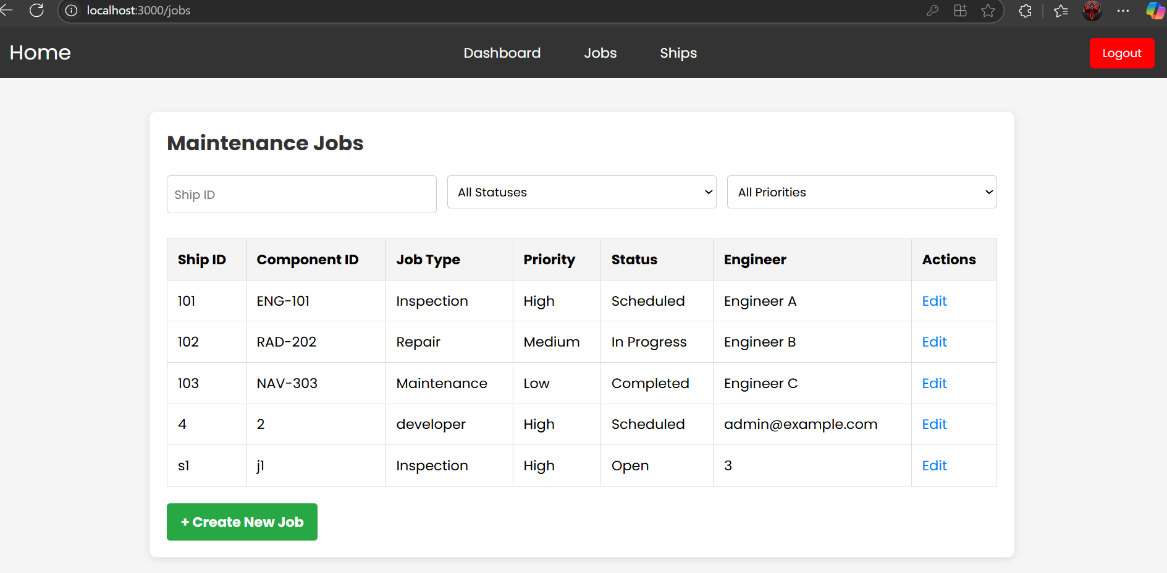
 an inspector can view only available ships and their status aslso IMO number,name,flag no,status and he can view ,edit and delete the ships based on data

 an engineer can view jobs; and an administrator can perform all operations.

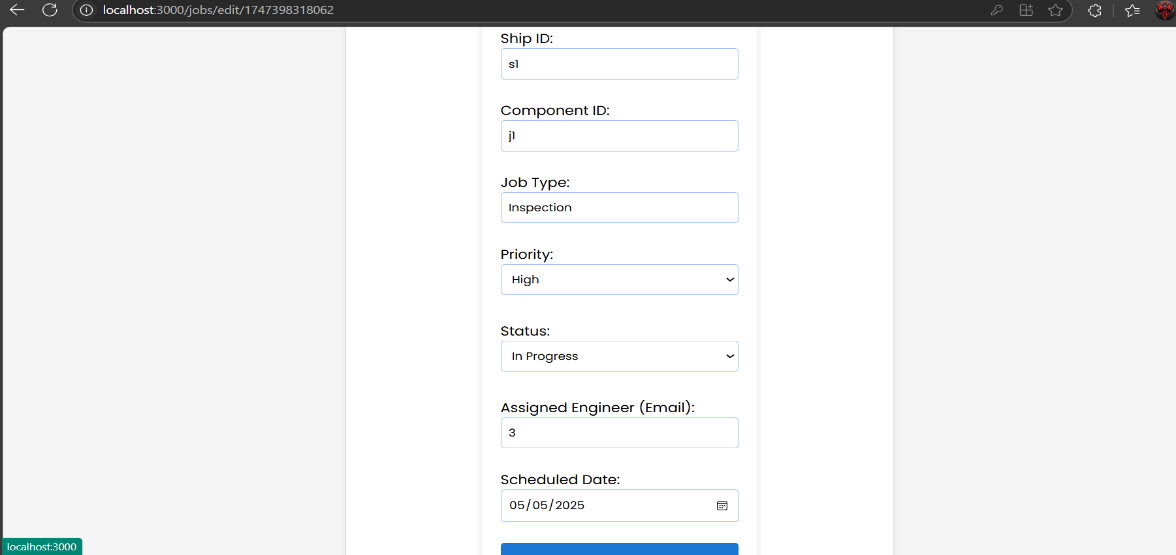
 Meanwhile, everyone can access the dashboard, which shows the total number of ships, overdue components, jobs in progress, and jobs completed.

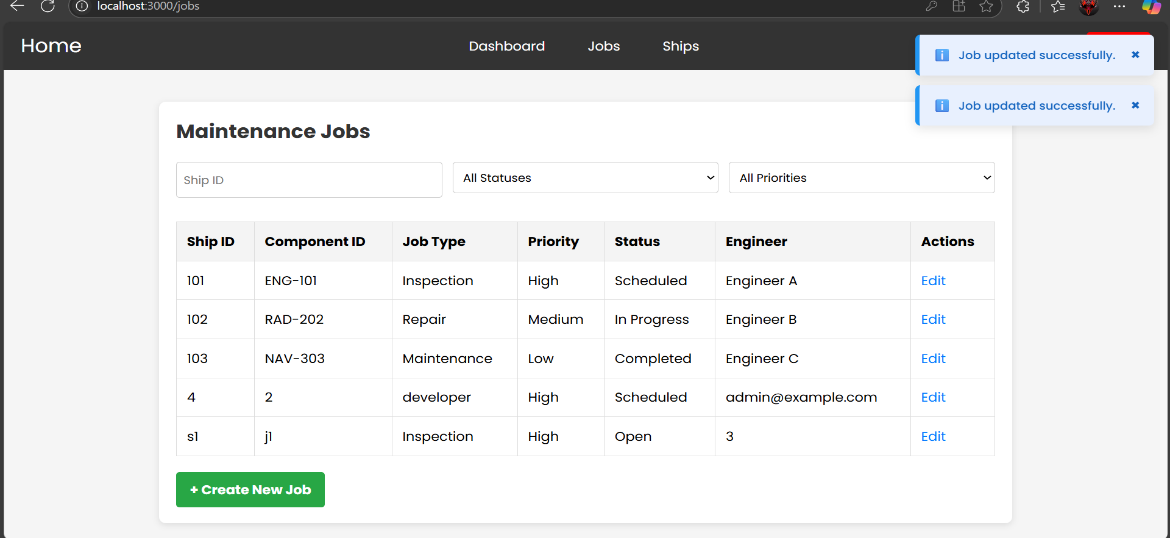
**For admin :**

after log in with the credentials the web page looks like this

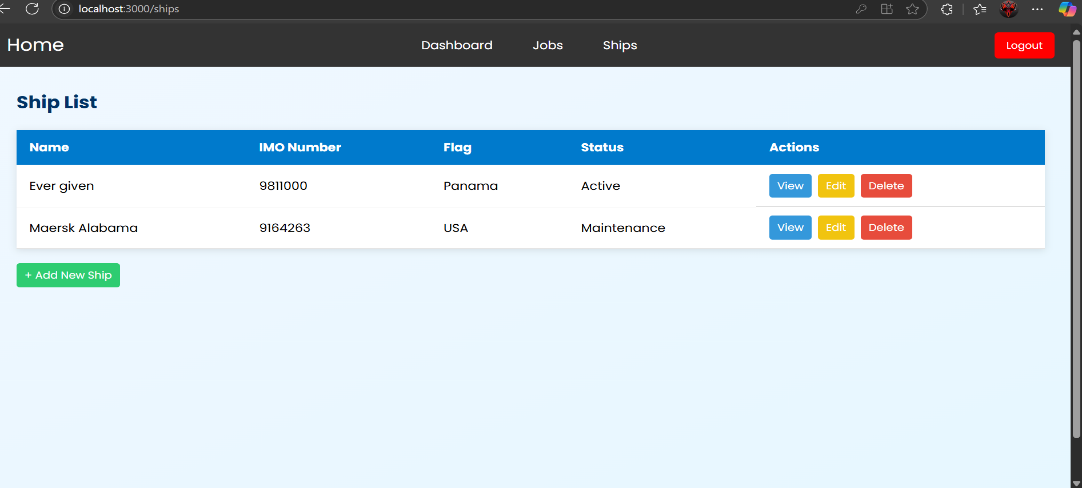
If an Admin wants to make any changes in the Job section, they can do so by clicking on the 'Jobs' tab, which provides full access to view, edit, create, or delete maintenance jobs 

After selecting "Edit Action," a form requiring modifications is displayed like this



we receive notifications regarding the job status if the administrator makes any changes to the job postings.

While the admin can only make changes to the ships tab, they are also responsible for ship monitoring and updating and ship page looked like this

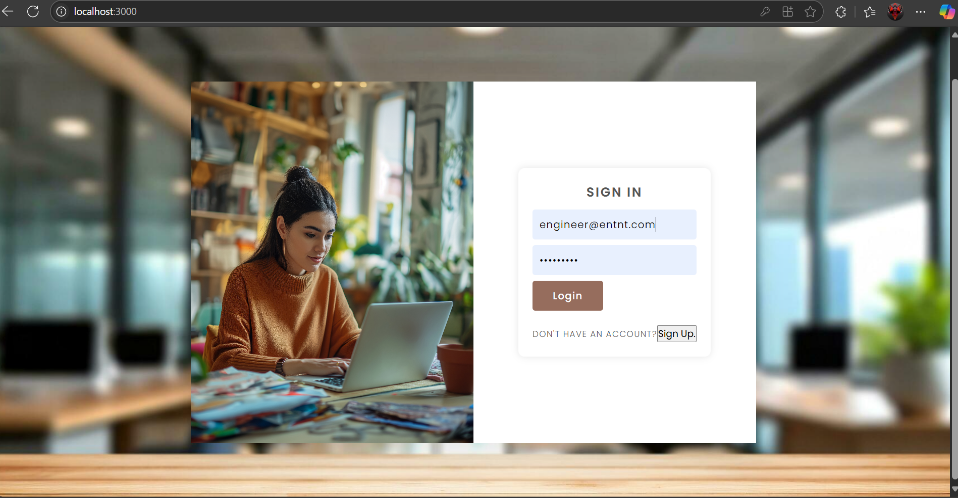


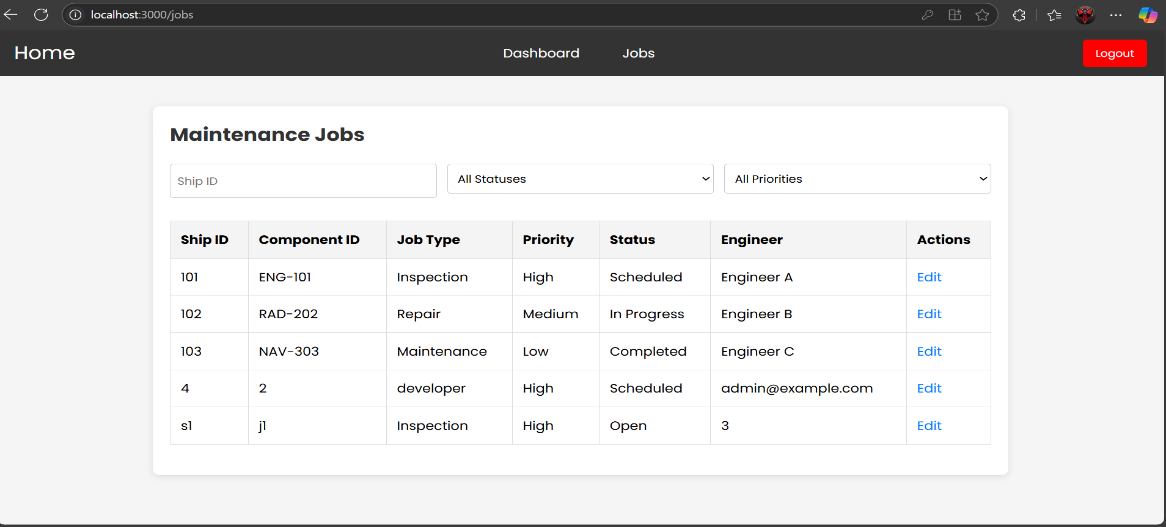
The purpose of the Ship List page is to present an overview of every ship that is presently under system management. It acts as a central overview, giving users instant access to each ship's important details, including name, IMO number, flag, and operational status.  
  
Additionally, this page serves as a starting point for managing specific ships. Users can remove ships from the system, change existing ship data, and access more detailed information by using the "View," "Edit," and "Delete" actions. Users can also add new ships to the management system by clicking the "+ Add New Ship" button.

By clicking on logout button admin can logout from page.

**For Engineer:**

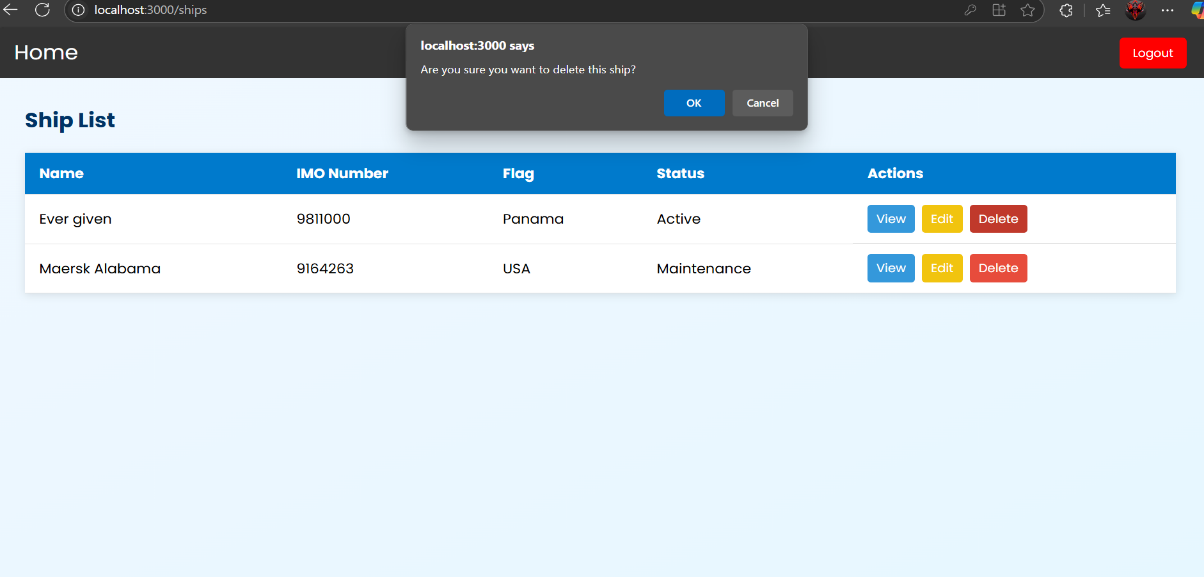
Actions Possible for a Engineer User:  
Enter their username or email address: In the corresponding field, enter their registered email address.  
Put in their password here: In the password field, type their password.  
Press the "Login" icon: Send in their login information for verification.  
Click the link that says "Sign Up": Go to the registration page, if your simulation has one.  
By comparing the entered credentials with your hardcoded user data stored in localStorage, this login page most likely simulates authentication based on the project requirements. After authentication is successful, the page will probably,To keep the user logged in, set up a session in localStorage.  
Depending on their role (Admin, Inspector, or Engineer), reroute the user to the main dashboard or another approved page.  
An essential first step in access control is this login page.



On the next page, engineers can access the dashboard and a jobs tab to see which jobs are available. "Maintenance Jobs" page provides a way to view and manage the maintenance tasks within your simulated system. To restrict engineers to only viewing job statuses, you'll need to implement role-based access control within your React components.

Once the engineer has viewed the available jobs, their next action will be to log out of the system.

**For the Inspector:**

****In the same way, the inspector is able to view the ships' status and make the changes in the ship list subsequently log out.

This React project delivers a simulated Ship Maintenance Dashboard with key features for managing ships, components, and jobs, all while persisting data locally and offering role-based access.