

CodeLogic – Payments Management System

1. Project Overview

This project is an Admin Dashboard Web Application designed to manage users, payments, and generate reports efficiently. It provides a secure interface for administrators to add, edit, and filter critical data in real-time. The dashboard is built using Vue.js 3, Pinia for state management, Vue Router for navigation, and JSON files for mock data.

2. Logic Walkthrough

2.1 User Management

- Imported user data from `user.json` and stored in a reactive variable using `ref()`.
- Displayed all users in a table with details: ID, Name, Email, Role, and Status.
- Added dynamic filters for User ID, Role (Admin/User/Manager), and Status (Active/Inactive).
- Only logged-in admins can edit user details.
- Non-logged-in users can view and search users but cannot make changes.
- Edit functionality implemented via `/edit-user/:userId` route and `EditUser` component.

Paymet Manager					Reports	Users	Payments	Login
User Management								
Search by User ID		All Roles		All Status				
ID	Name	Email	Role	Status	Edit User			
1	Admin	admin@gmail.com	Admin	Active	Edit			
2	Bob Smith	bob.smith@example.com	User	Inactive	Edit			
3	Charlie Brown	charlie.brown@example.com	Manager	Active	Edit			
4	Diana Prince	diana.prince@example.com	User	Active	Edit			
5	Ethan Hunt	ethan.hunt@example.com	Software Developer	Inactive	Edit			
6	Fiona Gallagher	fiona.gallagher@example.com	User	Active	Edit			
7	George Miller	george.miller@example.com	User	Inactive	Edit			

Edit User

Name:

Admin

Email:

admin@gmail.com

Role:

Admin

Status:

Active

Save Changes

Add User

Name:

Email:

Role:

Status:

Active

Add User

2.2 Payment Management

- Imported payment data from payment.json.
- Displayed payments with Payment ID, User, Amount, Status, and Date.
- Added search and filter options: Search by Payment ID or User, Filter by Status.
- Only logged-in admins can edit payments using /edit-payment/:paymentId.
- Non-logged-in users can view and search payments but cannot modify them.

Payments

Search by User or ID

All Status

dd/mm/yyyy

Payment ID	User	Amount	Status	Date
100000001	Alice Johnson	₹2,500	Completed	2025-09-10
100000002	Bob Smith	₹1,200	Pending	2025-09-11
100000003	Charlie Brown	₹3,200	Failed	2025-09-12
100000004	Diana Prince	₹1,800	Completed	2025-09-13
100000005	Ethan Hunt	₹2,700	Pending	2025-09-14
100000006	Fiona Gallagher	₹1,500	Completed	2025-09-15
100000007	George Miller	₹2,200	Failed	2025-09-16

Edit Payment

User:

Alice Johnson

Amount:

2500

Status:

Completed

Date:

10/09/2025

Save Changes

Add Payment

User:

Amount:

Status:

Completed

Date:

dd/mm/yyyy

Add Payment

2.3 Admin Authentication

- Implemented admin login using Pinia.

- isLoggedIn state tracks admin session.
- login(email, password) action authenticates admin.
- logout() action clears session.
- Navigation guards ensure only logged-in admins can access edit routes.
- Conditional rendering hides “Edit” buttons from non-admin users.

Login

Email:

Password:

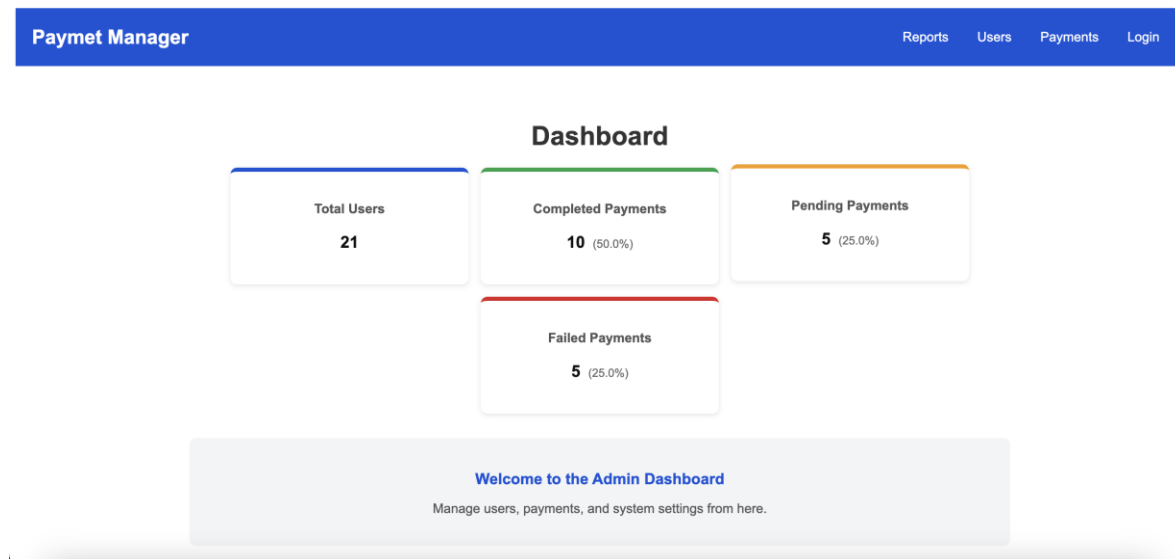
Login

2.4 Routing

- Used Vue Router for navigation.
- Public routes: /user, /payment, /
- Protected routes: /edit-user/:id, /edit-payment/:id
- Admin login: /admin-login(email: admin@gmail.com, password: Admin1234)
- Lazy-loaded components prevent circular import issues.

2.5 Dashboard & Reports

- Dashboard fetches user and payment data from JSON.
- Displays key statistics: Total users, active/inactive percentage, total payments, status-wise percentage.
- Provides a clear visual overview of the system.



3. Testing

- Component-level testing using **Vitest**.
- **Login Component:** Verified authentication logic, form validation, and routing after login/logout.
- **User Table Component:** Verified filtering, dynamic display, and conditional rendering based on admin login status.

4. GitHub Repository

<https://github.com/PremansuChakraborty/Frontend-mini-project.git>

5. Instructions to Run the Project

Prerequisites

- npm package manager

Steps

1. Clone the repository:

```
git clone https://github.com/PremansuChakraborty/Frontend-mini-project.git
```

2. Install dependencies:

```
npm install vue vue-router vuex
```

```
npm install --save-dev vitest @vue/test-utils jsdom
```

3. Run the development server:

```
npm run dev
```

4. For Testing:

`npm run test`

5. Open the browser and navigate to `http://localhost:5173(optional)`
6. Login using the default admin credentials in the `authStore` Vuex store. Only after login can the admin edit users and payments. Non-logged-in users can view and search users/payments but cannot make changes.

6. Notes

- Data is stored in JSON files (`user.json`, `payment.json`) for simplicity.
- CSS is scoped for each component to maintain modularity.
- The application is responsive and scalable for future enhancements.
- Admin-only routes are protected via navigation guards and conditional rendering.