**Plan and Overview Document**

**🔸 Project Name**

**AnimeVerse** – A dynamic anime content showcase with protected API access.

**🔸 Objective**

Build a secure Vue 3 frontend application that fetches random anime image data (with descriptions and passages) from a protected Express backend using JWT-based authentication. The data is managed using a Pinia store, displayed using Element Plus, and includes interactions like refreshing the image and viewing details.

**🔸 Tech Stack**

|  |  |
| --- | --- |
| **LAYER** | **TECHNOLOGY** |
| Frontend | Vue 3 (Composition API) |
| State Management | Pinia |
| UI Library | Element Plus |
| HTTP Client | Axios |
| Backend | Express |
| Auth | JWT (Json Web Token) |

**🔸 Features**

**✅ Frontend**

* **User Authentication (email-based)**:  
  Retrieves and stores JWT in localStorage using Axios.
* **Image Gallery via API**:  
  Fetches image data from backend /images endpoint using bearer token.
* **State Management**:  
  Stores and tracks image list, loading state, and current selection in Pinia.
* **Interactions**:
  + **Refresh** button to fetch a new image.
  + **Click on image** navigates to a detail page showing full description and passages.

**✅ Backend (Express)**

* /auth/login – POST → Issues token for valid user email.
* /images – GET → Protected route that returns full list of image data.
* Token validated using authenticateToken middleware.

**🔸 API Endpoints**

**POST /auth/login**

json

CopyEdit

Request Body: { "email": "user@example.com" }

Response: { "token": "<JWT\_TOKEN>" }

**GET /images**

* Requires Authorization: Bearer <token>
* Returns full list of anime image objects

**🔸 Pinia Store Design: useImageStore.ts**

**State:**

* token, imageData, loading, error, currentIndex

**Actions:**

* ensureToken()
* fetchImages()
* toggleImage()
* getImageByName()

**🔸 UI Layout**

**Home Page**

* Date & time at top
* User profile image at top-right
* Card with:
  + Anime image
  + Footer with refresh button
* Click on image → navigates to detail view

**Detail Page**

* Full image
* Description
* All passages (with heading + image + text)

**🔸 Error Handling**

* 403 Forbidden → triggers re-authentication (re-fetch token)
* Loading indicators shown during API calls
* Fallback UI for missing images or data

**🔸 Future Enhancements**

* Google OAuth or password-based login
* Add favorites / bookmarks for anime
* Backend MongoDB integration
* Search functionality