API Monetisation and usage requirements

Prerequisites:

The application should be built using Angular 15 or higher

UI Component Library: Angular Material should be utilized

Styling: Use Sass for better maintainability and reusability

Use Jest for unit testing

Collaboration is highly encouraged, but plagiarism isn’t. Work together but create something truly unique to you.

Functional Requirements:

User Registration and Login:

The app should have a user registration feature where users can create an account by providing necessary details such as name, email, and password.

Users should be able to log in securely using their registered email and password.

The app should provide appropriate error handling for incorrect login credentials and display relevant messages to the user.

Note: No API is required here for login or registration. Just allow the user through if validation is successful

Enable users to view and update their account information including contact details, billing preferences and API keys.

API discovery and documentation

API catalogue: Provide a searchable catalogue of browsable APIs with descriptions and usage guides

API filtering and sorting: Allow users to sort APIs based on categories, features, pricing plans or any relevant criteria

API usage monitoring and analytics

Display API usage metrics, including request volume, error rates, response times etc. Use charts and dummy data

Display Historical usage trends: provide historic usage data for analytics and reporting, allowing users to track API usage over time

API key management:

API Key generation: Allow users to generate and manage API keys

Key revocation: Provide the user the ability to revoke API keys that had already been generated.

Non-functional requirements:

Applications must be responsive for different screen sizes.

The application’s UI and UX are critical

The application should have localisation and internationalization

Write unit tests using Jest

Persona:

As a user, I’d like to be able to:

Browse some of the functionality your API could provide

Login to view API documentation and browse some of the different APIs you offer; Allow for searching, sorting and filtering.

Review the pricing list for the APIs

For a paying user,

allow them to generate API keys they can utilise.

Allow them to revoke API keys they’d previously generated

Allow them to review their billing details

Review their API utilisation and history

Version control:

In the Ebl-Developers-Playground, create a repo, you can name the repos as you’d like

Have atomic commits, and multiple branches for each feature or fix

Example sites:

https://platform.openai.com/

https://ai.google.dev/docs

https://www.meteomatics.com/