COMENIUS UNIVERSITY BRATISLAVA

FACULTY OF MATHEMATICS, PHYSICS, AND INFORMATICS

AI Content Generator

Final Report

Aidas Jankauskas
aidas.jankauskas@mif.stud.vu.lt

BRATISLAVA 2023

Contents

Repositories

Project is stored in 2 repositories:

- Frontend application: https://github.com/aidasjan/ai-content-generator-web
- Backend application: https://github.com/aidasjan/ai-content-generator-api

Both repositories are publicly available.

Public access

The application is deployed using Heroku cloud application platform. It is accessible under the following URL: https://aicongen.herokuapp.com/

Development environment

Development environment consists of 3 infrastructure elements:

- Frontend application
  - Running on port 3000 by default
- Backend application
  - Running on port 8080 by default
- MongoDB database
  - Running on port 27017 by default

In order to setup the development environment, the following steps must be done:

1. MongoDB must be installed on the system. The following documentation can be followed in order to do it: https://www.mongodb.com/docs/manual/installation/

2. Environment variables must be set for backend application (.env file can be used):
    a. PORT – port that backend application uses to serve requests.
    b. DB_CONNECTION_STRING – MongoDB database connection string.
    c. JWT_SECRET – secret string used to sign JWT tokens.
    d. JWT_VALIDITY_MINUTES – number of minutes for JWT token validity.
    e. OPEN_AI_TOKEN – secret token for OpenAI services.
    f. OPEN_AI_URL – OpenAI API URL.
    g. OPEN_AI_MODEL – OpenAI model to be used for content generation.
    h. OPEN_AI_TEMPERATURE – OpenAI temperature property for content.
    i. OPEN_AI_MAX_TOKENS – number of maximum OpenAI tokens to be used for content generation requests.
3. Environment variables must be set for frontend application (.env file can be used):
    a. REACT_APP_API_URL – URL of backend application.
4. Frontend and backend application dependencies must be installed using: `npm install`
5. Frontend and backend application can be started by following commands:
    a. Frontend application: `npm start`
    b. Backend application: `npm run dev`

Implementation progress

- All features according to the specification are implemented and usable.
- Progress since beta version:
    - Data management and security updates.
    - Application is tested manually.
    - User experience improvements are implemented.

Problems

- Maintenance cost of the software. Every OpenAI request uses tokens that are billed. Deployment to Heroku platform is also billed. In order to resolve the unexpected usage problem, a spending cap is set on both platforms. For the current version, the budget should not exceed 10 EUR per month.
- Uncertainty regarding the code quality. The experience level of using Express.js is limited, so it is uncertain if the code is implemented in the right way to ensure future scalability and maintainability, prevent unexpected memory leaks. This problem was partially resolved after consultation with a specialist that has more experience with the framework.

Retrospective

If the project would be started again, the following things might be done differently:

- Backend framework selection. Currently the chosen backend framework is Express.js. A framework that is more focused on OOP principles and more advanced design patterns (such as dependency injection) might be used in order to ensure better maintenance in the long run. However, it is not a major issue since the project is not expected to be scaled and extensively extended in future.
- Application architecture. Currently the architecture is monolithic, which means that the application is hardly scalable for greater amounts of users. Microservice architecture might be used to ensure higher scalability. However, it is not a major issue since the project is not expected to be scaled.

Overall, the project is implemented well and has no major issues. The most usable feature is integration with OpenAI API to create AI generated content. Other features, such as publishing in a blog are implemented well but are less relevant for a casual user. Multiple users have tested the application and were happy with the features that it provides.