

Technologies & Tools

Development Environment

Since our allocated project is a continuation of a team from the previous semester, our team has decided to use the technology stack that the previous semester's team built the existing codebase with.

The existing project uses a micro-services architecture to arrange the web application as a set of loosely-coupled services, including a backend server, frontend client, and database.

The specific technologies used for each of these microservices / module are detailed below.

Backend

- Instead of using FlaskAPI like other teams from the previous semesters, we have decided to use FastAPI (<https://fastapi.tiangolo.com/>) because we want to try something new and we all know that FastAPI is the state-of-the-art.
- Swagger (<https://swagger.io/>) to generate API documents.

Database

- MySQL (<https://www.mysql.com/>).

Frontend

- ReactJS (<https://reactjs.org/>). We use the latest programming style of ReactJS that is building functional components with hooks instead of building class components.
- JavaScript (ES6).
- HTML5/CSS3.
- Antd (<https://ant.design/>). Like the previous team, we use this UI framework to speed up the UI development.
- Webpack.

Deployment/Hosting

The application is deployed on an AWS EC2 instance, and uses a MySQL database hosted on AWS RDS. The hosting environment has been provided by the client.

- Docker
- Amazon Web Services (S3, Amplify, EC2, RDS).

Source/Version Control

- GitHub

Testing

- Frontend Testing Framework: Jest + React Testing Library (<https://jestjs.io/docs/tutorial-react>).
- Backend Testing Framework: pytest (<https://pytest.org/>).
- Integration Testing: Selenium.

Kanban Board

- Trello.

Mockups/Prototypes

- Figma.

Communication

- Slack.