EE 547 – Applied and Cloud Computing

Project Proposal

©Group 3

November 18, 2021

1 Project Title

Programming Q&A Platform

2 Team Members

Haoyu Guo, Ziyou Geng, Zhenyang Li

3 Summary and Description

In this project, we propose to build a coding forum website like Stack Overflow. Users can post all kinds of questions about programming to the platform and can also answer questions from other users. Specifically, the website will have a question publishing module, a question list module and a question content module with functionalities for modifying content, deleting content and adding comments. The platform will also have a complete user management system, including user registration and login, authority management, personal information modification and profile picture upload. After development, We will deploy the whole project to AWS.

4 Proposed Architecture

Our project will consist of a client-side (Front-end) and a server-side (Back-end), plus databases and publishing services. Both the Front-end and the Back-end will adopt MVC pattern.
The main technology stack is Vue.js + Node.js with Express + MongoDB with Mongoose. The technologies and libraries used at the Front-end and Back-end are described in detail below.
For the Front-end part, we will utilize Vue.js with Vue Router, VueX and Ant-Design-Vue UI library. The project will be initialized by Vue CLI. On this basis, text editing will be implemented with rich text editor module, user login and authority management will be realized based on JWT and all requests will be sent with Axios.
For the Back-end part, We will utilize Express and Mongoose. RESTful API will be adopted and CRUD operations will be conducted by Mongoose. We will also implement user authentication, file uploading and downloading, Real-time notification and any other functionalities we find useful.

Login Page Accept usernames and corresponding password, redirecting to **Home Page** if successfully verified. Return error message otherwise.

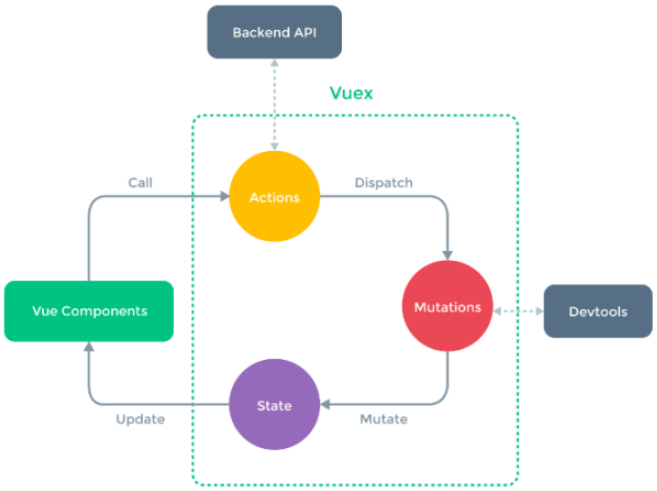

Figure 1: Vue dataflow

Sign-up Register as a user. Gather basic info such as email, unique username and password. (Probably more detailed information to gather user profile.)

Home Page Page with recommended questions, news and recent updates as the main interface

Question Page Full content of a specific question, let users answer questions and add comments.

Content Editing Page with rich text editor, enable users to edit and post questions.

User Profile Show user information and overview of the questions they have posted and answered.

We will use a local MongoDB database server to store user data and an AWS S3 to store the original and processed files in the cloud. This will simplify the async processing pipeline. We will either use a task server to process content and report to the Back-end or investigate using AWS Amplify.

5 Datasources and Additional APIs

Users can post their own questions and the answers will be posted by other users. Data related to questions and answers will be stored in databases.

6 References, Tutorials, Codebases, Documentation, and Libraries

- [1] wangeditor: a useful rich text editor module. https://github.com/awamwang/vue-wangeditor-awesome
- [2] Design and Implementation of a Vue.js-Based College Teaching System
- [3] Vue 3.0 official documentation: https://v3.vuejs.org/api/

7 Estimated Compute Needs

We will create a new private VPC in the AWS cloud. We anticipate a single t2.small server will suffice. We will use AWS Simple Queue Service (SQS) and Simple Notifiation Service (SNS) to control the async processing. We don't anticipate any unusual hardware or compute needs.

8 Team Roles

The following is the rough breakdown of roles and responsibilities we plan for our team:

- Haoyu Guo: Primary: Back-end develop and API devise, MongoDB resolvers and database access. Secondary: assist with some layout elements.
- Ziyou Geng: Primary: Front-end design. Secondary: Assist in API devise, ensure that client-side requests and Back-end API are compatible.
- Zhenyang Li: Primary: Design test cases for the server-side, including Mocha and write JavaScript workflow code like CirclCI to implement continuous integration and delivery. Secondary: host the website in AWS

All team members will work on the final presentation, slides, and report.