

Symphony of Syntax

Symphony of Syntax is an online platform where budding programmers and developers can hone their programming skills. The platform offers a seamless experience for users to solve coding problems, receive instant feedback, and track their progress. It supports multiple programming languages, provides a robust code execution engine, and features comprehensive problem management and evaluation systems.

Frontend: (ReactJS)

- **Login / Register Page :**
Here the new user will register himself / herself with name, email, password and username(preferably email). Existing users will use their username and password to login.
- **Home Page :**
The user will be exposed to different problems here. They will choose the problems they want to solve. Also, it will show the problems that they have solved.
- **IDE Page:**
The user will get the complete information about the problem. The problem statement, input and output form, constraints and example. Users will also choose the language in which they will code and space for code will be given. They can either run their custom test case input or can submit to check the accuracy of code.

BackEnd: (Django):

- Various django restful APIs will be used for smooth manipulation and access to the database and some other operations.
- Some tentative APIs:
 - ❖ /login/
 - ❖ /register/
 - ❖ /prob_lists/
 - ❖ /prob_id/

- ❖ /submit/
- ❖ /run/
- ❖ /logout/

Database(PostgreSQL/MongoDB):

- The Database will store the following details of the users:
 1. Name of User
 2. User id
 3. Username (tentative)
 4. Mobile No. (tentative)
 5. Email
 6. Password (Encrypted form)
 7. Number of Questions solved (tentative)
- Following will be stored for problems:
 1. Problem id
 2. Inputs
 3. Outputs
 4. Test cases
 5. Problem title
 6. Statement
 7. Total accurate submissions
- Following Data will be stored for code:
 1. Submit Id
 2. User Id
 3. Code
 4. Result

Deployment:

Use of Docker and AWS will be used to deploy the online judge website. Docker will be used to deploy the website and AWS will be used to use databases on cloud.

Vulnerabilities:

- DDOS attack
- Unauthorised modification of test cases
- Unauthorised modification of other user's data and codes

- Submission of Malicious Code

Security Measures:

- Limiting rate of running codes
- Custom Isolation
- Writing safer code
- Limiting permissions and user access
- Limiting number of submissions in given time