Gator Help Portal
CSC 890 - Team 03

Developers:
Prathiba Ramesh
Anudeep Katukojwala

Introduction or App Summary

- This app allows SFSU CS department students to get **instant advising** on various subject matters that help them succeed in their field.
- This process will ease the work and **reduce the effort** of both department faculties and the students.
- One stop solution for all academic needs of a student, as they get details about **important dates**, procedure and advising.
- Ease of accessing multiple data, as browsing of data from multiple websites to get targeted results can be avoided.
- Student's and Faculty's **wait time is reduced**, as they need not wait to ask their queries on a one-on-one session. Instead, our app can provide all the details in just one click.

Priority Features

Account Management:

- Users shall be able to register their accounts with the app.
- User shall be able to log into (with registered Cred) and log out of their accounts to gain access and deny access from the app.
- Users shall be able to manipulate their profiles to give their basic information, and general information on their academics which shall be used to provide suggestions.
- Application shall require users to **verify their email address** upon registration, **password** should be of **8-13 characters**.
- User's personal data shall be **hashed/encrypted** from public access for additional security.
- User shall be able to add their role on registering to the page. **Roles** can be General User or Administrator.

Priority Features

Deadline/ Important Date Information:

- Users shall be able to ask any questions related to academic deadlines and priority dates, and get back appropriate response from AI
- User shall be able to ask the same question in any variations, and they should still get the correct desired answer.
- User shall be able to give **Feedback** on whether they are satisfied or not, based on the answer they received. This is optional.
- User shall be able to give **Rating** (star based) based on their level of satisfaction with the answer. This is optional.User shall be able to give **additional comments** as well.
- User can repeatedly ask the same question and the AI should give back exact same answer.

Priority Features

Dashboard/ History of Data:

- **User** should be able to visit the **history of questions and answers** that they have asked and received so far.
- **Authorized** User shall be able to view the entire **list of users and all the questions** they have asked so far, along with the answers they have received.
- Authorized Users shall be able to see the **feedback** provided by users for each question and answer they have received.
- User and Authorized User shall be able to **filter** the user data based on feedback their student id.
- User and Authorized User shall be able to **sort** the user data based on student id, feedback they have provided or rating.
- User and Authorized User shall be able to select the number of data they want to view in single page view.

Application Demo

Web url:

http://sfsuhelp.eastus.cloudapp.azure.com/

Or

http://20.120.20.37/

Software Architecture

OpenAI Algorithm:

- We are using the GPT-3 model (developed by OpenAI) as our API to generate answers (completions) based on the question provided (prompt).
- The necessary data is fetched through Post method to API call https://api.openai.com/v1/completions
- *Model:* "Davinci" ; *Fine-tuned Model:* davinci: "ft-personal-2022-10-30-00-57-34"
- Data Source
- Fine Tuned Dataset
- Test Data
- API key → Ignored on Git commit for security purpose.

Software Architecture

Tech Stack:

- ***Cloud server:*** Google cloud → Microsoft Azure
- ***Operating system and version:*** Debian 11
- ***Database:*** MySQL
- ***Web server:*** Node.js
- ***Web application framework:*** Express
- ***Server-side Language:*** JavaScript
- ***Client-side Framework:*** React.js
- ***IDE:*** Microsoft Visual studio
- ***AI stack:*** Open AI → Davinci Model

Software Architecture

Database Management:

- Azure MySQL instance:
  - Host : *20.228.204.41*
  - UserName: *csc890*
  - Password: *Sfsuhelp890*
  - DB Name: *cschelp*
- Connection details:
  - SSH private key
  - ssh -i <private key path> azureuser@20.228.204.41
- DB entities and attributes:
  - *users:* emailAddress, userName, password, studentID (PK), currentSemester, userRole
  - *userquestionlog:* studentID (FK ref users), userName, userRole, question, answer, feedback, loggedDate

Software Architecture

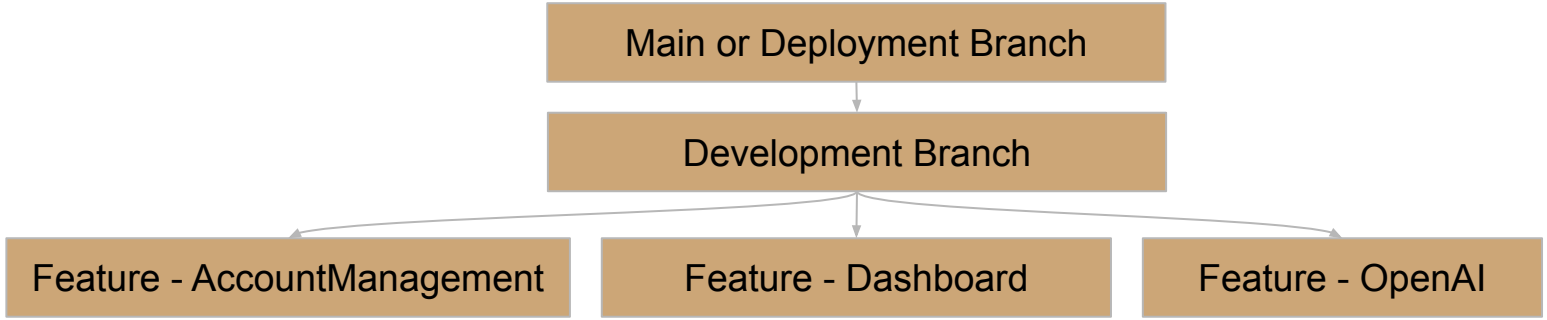
Azure Cloud for deployment

- Connection details:
  - SSH private key
  - ssh -i <private key path> csc890Team@20.120.20.37

Github Management

Architecture:

- Github Link → https://github.com/PrathibaRamesh/CSC890-03-Fa22-Team02
- Github Branches and hierarchy:



- 10 closed Pull requests and Code review
  - Created when Code merged to deployment branch.

Github Management

Coding Style:

- The code is maintained in such a way that it is self-explanatory.
- The Naming convention in the code for functions and variables are well maintained as per the standard and is consistent throughout our project file.
- The code is properly indented for better understanding
- Each of the functions are well documented as to know what exactly it does.
- Comments are provided for each of the critical functions in the code.

Github Management

Code Review:

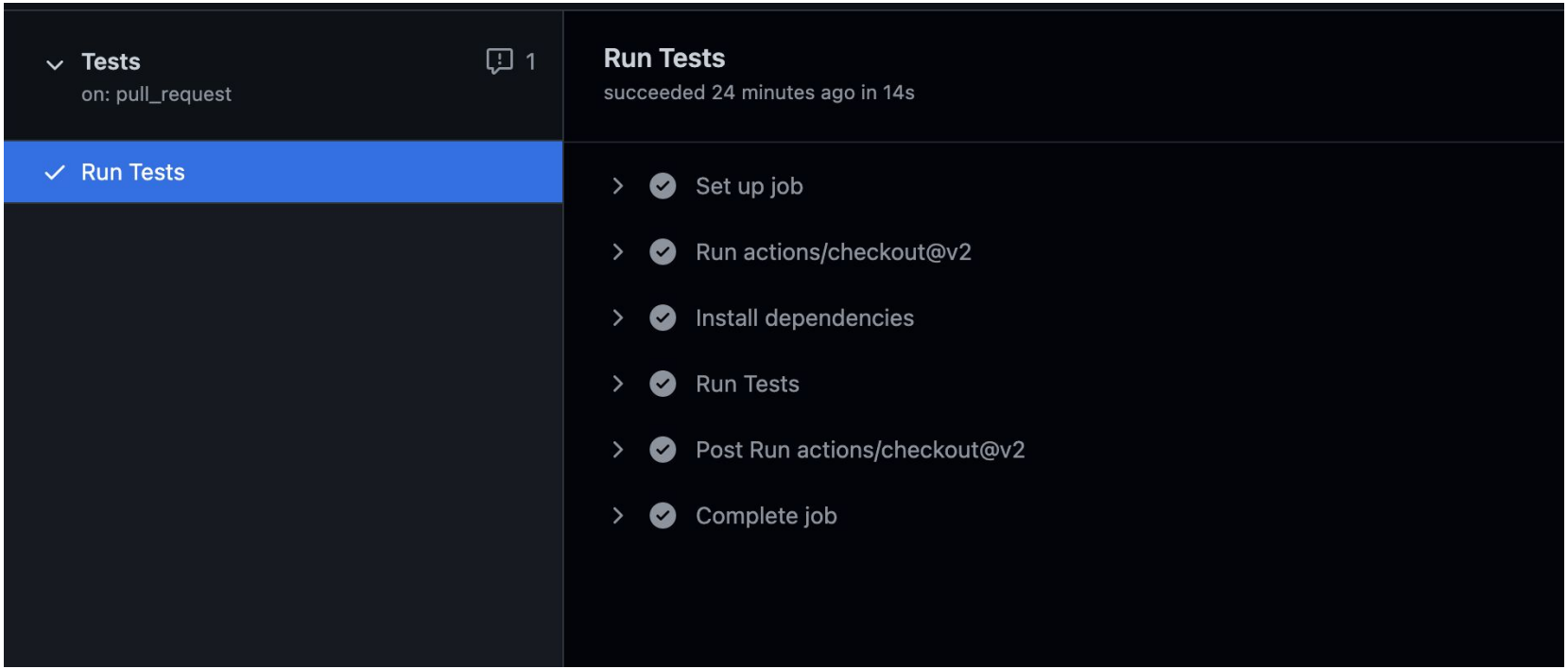
Once the code has been developed by the frontend team, the backend team was assigned for the code review and vice versa.

Below are the Git Pull request URL's that we fetched from GitHub for each code review requests and approvals,

- https://github.com/PrathibaRamesh/CSC890-03-Fa22-Team02/pull/11
- https://github.com/PrathibaRamesh/CSC890-03-Fa22-Team02/pull/12
- https://github.com/PrathibaRamesh/CSC890-03-Fa22-Team02/pull/16
- https://github.com/PrathibaRamesh/CSC890-03-Fa22-Team02/pull/19

Github Actions

Github Actions- Continuous Integration (CI):



Test Coverage - *Unit Test*

```
PASS ./index.test.js
  Testing validation functions
    ✓ Test for valid email (2 ms)
    ✓ Test for valid password
  register route
    ✓ testing email validation (97 ms)
    ✓ testing password validation (9 ms)
    ✓ testing successfull registration (331 ms)
  login route
    ✓ testing login verification (97 ms)
    ✓ testing login icorrect part of the code (89 ms)
  getFeedbackData route
    ✓ testing data if userRole is General User (92 ms)
    ✓ testing data for wrong user when userRole is General User (92 ms)
    ✓ testing data if userRole is Administrator (90 ms)
  Testing question register route
    ✓ testing if the request is rejected when all parameters are not sent (15 ms)
    ✓ testing insert data (98 ms)

----------------------|---------|----------|---------|---------|-------------------------
File                  | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
----------------------|---------|----------|---------|---------|-------------------------
All files             |   92.85 |    78.78 |    92.3 |   92.85 |
 connectionDetails.js |     100 |      100 |     100 |     100 |
 helperFunc.js        |     100 |      100 |     100 |     100 |
 index.js             |   91.86 |    78.78 |    90.9 |   91.86 | 75,100,124,141,148,179,193
----------------------|---------|----------|---------|---------|-------------------------
Test Suites: 1 passed, 1 total
Tests:       12 passed, 12 total
Snapshots:   0 total
Time:        2.271 s
Ran all test suites.
```

Test Coverage - *Integration Test*

CSC890-Team02-Integration Testing

File Edit View Insert Format Data Tools Extensions Help

100% View only

A1:B1 fx Test Case Description

1	**Test Case Description**		Test the Register New User Functionality in SFSU CSC Help App								
2	**Created By**		Prathiba	**Reviewed By**		Anudeep		**Version**		1.0.0	
3											
4	**Tester's Name**		Anudeep	**Date Tested**			20-Nov-2022	**Test Case (Pass/Fail/Not Executed)**		Pass	
5											
6	**S #**	**Prerequisites:**				**S #**	**Test Data**				
7	1	Access to URL http://20.120.20.37/									
8	2										
9											
10											
11	**Test Case ID**		**TC_001**	**Test Scenario**	Verify on entering valid input entries and user should be able to register						
12											
13	**Step #**	**Step Details**		**Expected Results**		**Actual Results**			**Pass / Fail / Not executed / Suspended**		
14											
15	1	Navigate to Above URL		Site should open		As Expected			Pass		
16	2	Click on the "New User? SignUp" button		The Register page must open up		As Expected			Pass		
17	3	Enter a valid Full name, email address, student id, role in appropriate text field		Data can be entered		As Expected			Pass		
18	4	Enter a password in appropriate text field		Data can be entered		As Expected			Pass		

Team Management

- A weekly meeting over zoom to exchange ideas, helped us to be on track in this project

- For every milestone, we have divided the work, went back to read about them and discuss them in our weekly meeting.

- It was hard to coordinate sometimes since we have other classes but we made it work by being consistent to our weekly meetings.

Challenges and Risks

- We had issues with Google Cloud where our $100 credit got deducted without any notification.

- We had then switched in Azure, but at the last moment just a day before M5 presentation our free credit expired and has to switch to pay as you go subscription.

- Apart from the issues with cloud provider, we had issues with fine tuning the model in the initial weeks.

- After a good amount of research, the OpenAI api worked as intended in our project.

Thank You.