SJSU Student Assistant System

Aviral Kumar
Sai Teja Desu

Varun Shah
Jay Desai

***Abstract*— There are around 32,000 students at San Jose State University and all of these students are enrolled in huge variety of courses which include technical as well as non-technical. Students attending the university come from different education background, culture, skills. Often when the students take up a course there are certain issues that face. On one hand we have these set of students who face these issues while on the other hand there are a set of students who have already faced these issues earlier and they can be of help to the current student in need of it. Currently there isn't any platform that brings together these set of people together. SJSU Student Assistant System aims to solve the above problem. It brings together the set of students that need help and another set of students who can help. The student who provides some help, can also in turn require some help from other students in the community. Every student in the system will have his own profile, course, skills listed which the system can use to map to students that require help.**

I. Introduction

The current problem of disconnection between 32000 students is what SJSU Student Assistant System is basically aimed to solve. The main idea behind this system is provide students a platform where they can come, raise the issues they have, meet students from their course work who in turn provide them with the solution or possibly some reference which can be useful for them. The goal of the system is to create a community of students where they can discuss and find solution to their issues and problems.

Student Assistant System has a simple and easy to use interface which allows the users to raise an issue, check who is ready to solve it. The system has some basic requirement which the user needs to fulfill before he can go ahead and use it. The student has to sign up with his details which include mandatory SJSU ID without which the user won't be able to use the system as it is meant only for students of the university.

As soon as the student registers into the system he will have the options of editing his basic profile information. In the profile section he can specify the skill, course work he has already taken before. The user can save the information and then later on edit this if he wants to add some more information.

There is a section where the student can go and look up the currently open issues in areas which he has mentioned in his skill set. This page will list all the issues raised by different students. The student will have the options of viewing the issue and providing help to the student who has raised the issue.

The student who has logged in has also the option to raise issue in a particular area or subject on the "Raise Issue" page. The user can raise the issue and then wait for a user who is ready to provide help.

II. Student Sign Up Page

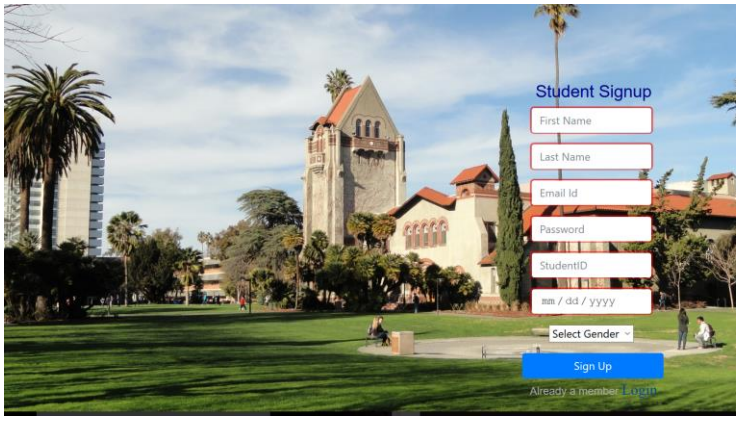


Figure 1.1 Student Sign Up

Shown above in the figure is the student sign up page. This is the first step towards using the system. The user has to enter below mandatory details before he can use the system :

1. First Name
2. Last Name
3. Email Address
4. Password.
5. Student ID.
6. Date Of Birth
7. Gender.

Once the user enters the above details, he is successfully registered into the system. Using the credentials above the user can now log in to the system as shown in the next figure.

III. STUDENT LOGIN PAGE

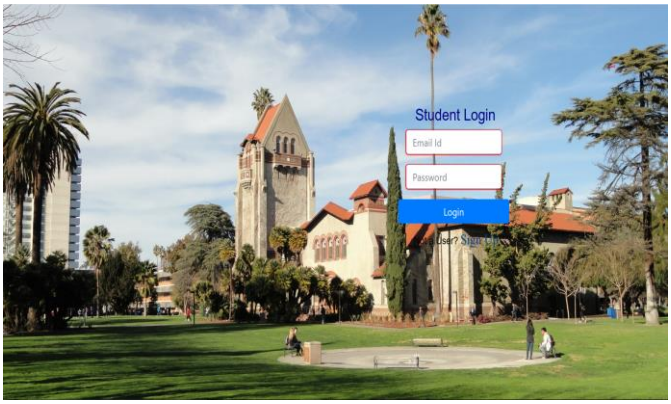


Figure 2.1 Student Log In

Above is the page which the student can use to log in to the system. After successfully registering the user needs to enter the email and password on this page to log in.

IV. STUDENT PROFILE PAGE

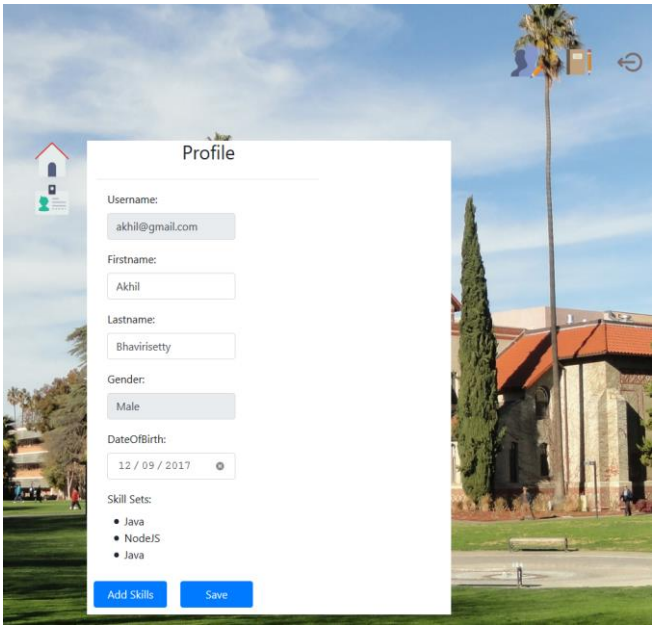


Figure 2.2 Student Profile

Above is the Student Profile page where the student has the option of editing his basic profile information like the skill set. He can update his or her skill set , courses taken up. The value in this field will be used to display the currently open issues in this skill set.

V. OPEN ISSUES PAGE

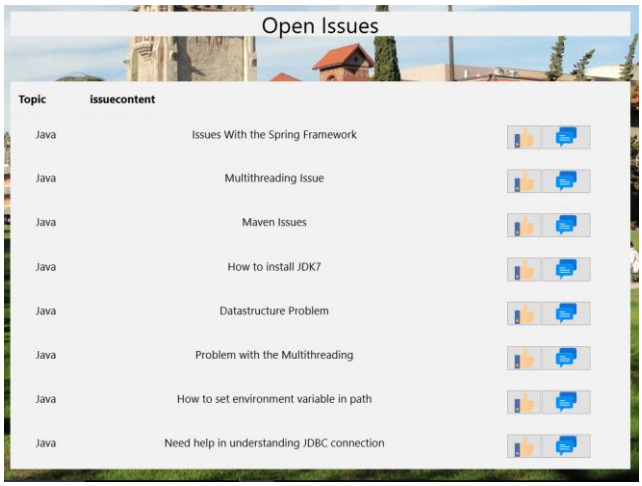


Figure 2.3 Open Issues Page

This is the place where the student can find all the currently open issues in the area which he has mentioned in his skill set on the user profile page. The topic name of the issue and the content of the issue along with it is displayed here. The user can scroll through the issues displayed and can provide help to the concerned user if he wants to the student who has raised the issue.

VI Recent Activity

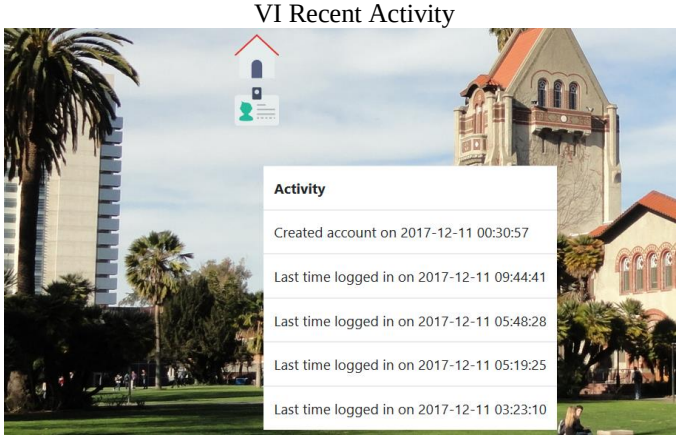


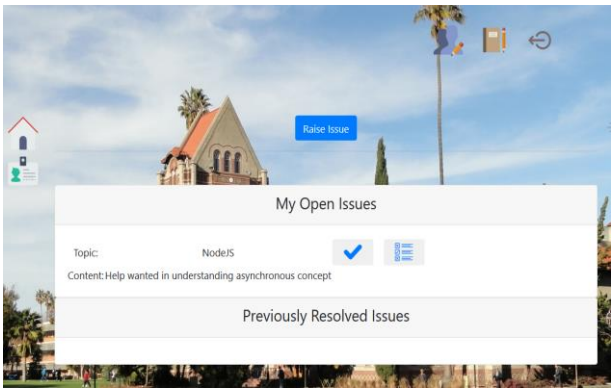
Figure 2.4 Recent Activity

Above figure shows the page where the student can see his recent activities. As shown in the figure above it displays two activities that were performed by the student:

1. Account Creation
2. Last login details.

It displays the date and time when the user created the SJSU Student Assistant System Account and also the date and time when the user last logged in to the system.

VI RAISE ISSUE



server-side and networking applications. It enables event driven programming to the web servers.

Figure 3.1 Raise Issue Page

The screenshot above displays the page where the student can Raise his or her issues related to a particular subject or skill. The user does this by clicking on the "Raise Issue". Apart from Raising the issue on this page the user also sees the list of open Issues raised by him and also the list of issues resolved. This Functionality helps the user in keeping track of the several issues Raised by him and out of those how many are resolved and how Many are still pending.

VII Front-End Technology

The Front-end of the "SJSU Student Assistant System" is designed using React. The reason behind choosing "React" as the front end was its component based approach towards the UI. It helped in dividing the UI into a set of components which were easy to update. Each component decides how it should be rendered. The components designed can be re-used later. It also gives an enhanced user experience which is very important.

VIII Back-End Technology

The Back-End of the "SJSU Student Assistant System" is built in Node.js. It is an open-source cross-platform built on top of the chromes JavaScript runtime for fast and scalable

XII Flow Diagram

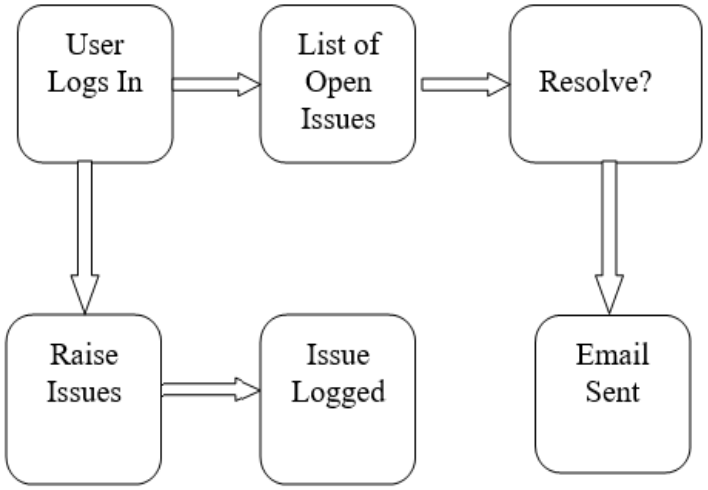


Figure 3.2 Application Flow

Above diagram describes the application flow of "SJSU Student Assistant System". It is divided into six blocks. In the first block the user logs in and the home page is displayed. On the home page the student can see the list of open issues according to the skill he has mentioned in in his skill set in the User's Profile page. The user has the option of changing his skill on the user profile page.

Issues related to the skills mentioned on the profile page will be displayed on the user's home page. Moving further user has the option of either resolving a particular issue or raising an issue himself. User can select a specific issue from the list of issues displayed And then click on the "Resolve" button. As soon as the user clicks on the "Resolve" button an email is sent to the concerned user along with the details of student who has agreed to help.

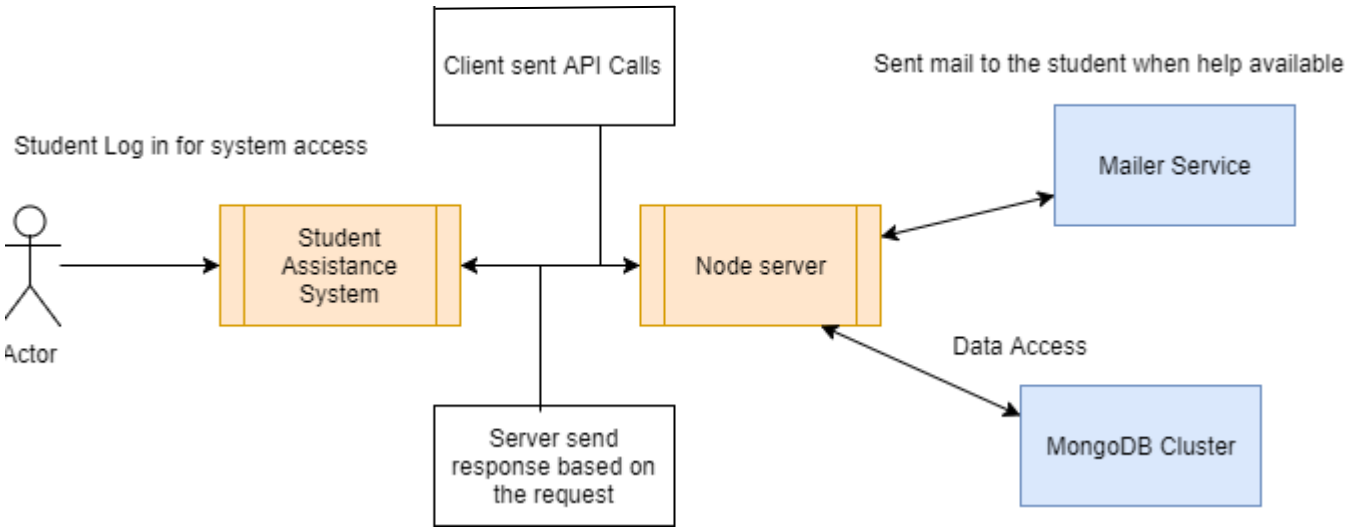


Figure 4.1 Architecture Diagram

XIII. Architecture Description

Displayed above is the architecture diagram of the "SJSU Student Assistant System". Architecture consists of several modules.

Starting from the left they are: Actor which uses the system, student assistant system, Node server, MongoDB cluster, Mailer Service. Actor interacts with the student assistant system by accessing the UI of Student Assistant system. The client which is built using "React" is the frontend which the user is able to see. Using the front-end the user makes all the request to the server. Each request sent from the front-end hits an API at the back-end which then processes the request and sends a response to the user. The request is sent to the user in the form of URL. At the backend there is a Node server running which is ready to take up the requests from the client. As soon as the user makes a request say for example raising an issue, the corresponding API at the backend is called to process the request. The corresponding API will interact with the MongoDB which is running at another port. The API will add the issue raised into the DB and then send a response to the user that issue has been raise successfully.

At the backend we have an API which is used to send mail to the user. As soon as the user agrees to resolve an issue of a particular user , then a mail along with the description of the user who has agreed to resolve the issue is sent to the concerned user. There are multiple reasons to use MongoDB as the database for this application rather than relational database such as MySQL. Mongo DB's flexible data model presents a superset of other database models. It makes data to be presented as key-value pairs and flat, table-like structures, through to rich documents. With an expressive query language, documents can be queried in many ways – from simple lookups to creating sophisticated processing pipelines for data analytics and transformations, through to faceted search, JOINs and graph traversals. MongoDB stores data in a binary representation called BSON (Binary JSON). The BSON encoding extends the popular JSON (JavaScript Object Notation) representation to include additional types such as int, long, date, floating point, and decimal128. BSON documents contain one or more fields, and each field contains a value of a specific data type, including arrays, binary data and sub-documents. In short , relatively high speed of document retrieval in MongoDB and high performance makes the right choice for SJSU Student Assistant System which requires less joins among tables but requires faster retrieval of information to enhance user experience and provide them a fast environment.

The Front-end of the "SJSU Student Assistant System" is designed using React. The reason behind choosing "React" as the front end was its component based approach towards the UI. It helped in dividing the UI into a set of components which were easy to update. Each component decides how it should be rendered. The components designed can be re-used later. It also gives an enhanced user experience which is very important. The Back-End of the "SJSU Student Assistant System" is built in Node.js. Node.js is an open-source , cross-platform built on top of the chrome's

JavaScript runtime for fast and scalable server-side and networking applications. It also enables event driven programming.

XIV.Conclusions

SJSU Student Assistant Website aims in connecting students who are in need help and students who are willing to help fellow students based on their expertise and experiences in various fields.

XVI.References

1. https://github.com/facebookincubator/create-react-app
2. https://getbootstrap.com/
3. https://docs.docker.com/engine/userguide/
4. https://redux.js.org/docs/introduction/
5. https://docs.atlas.mongodb.com/
6. https://aws.amazon.com/getting-started/tutorials/deploy-docker-containers/