1. INTRODUCTION

1.1 General Introduction

Managing data in a very large and long-established institution is a tedious job. As the size of the institution keeps on flourishing, maintaining annual data of the institution in a format that is readily accessible is critical to the institution. RIT Data-Center records all data regarding the institution, various committees and departments of the institution, achievements and activities conducted by various departments of the institution, admission and scholarship details of the students, faculty details of each department of the institution and so on in a structured form that can be read, queried and updated in an effortless manner. The system provides various levels of authentication and authorization to the members of the institution to view and update data in a central database. The primary function of the system is to generate dynamic reports for every department and the institution in the specified format for the given requirements. The system also encourages transparency in holding the data as various users with required authority can directly monitor data flowing in and out of the system. Moreover, unlike conventional methods, the system will be available all the time. Also, it reduces the time taken for data collection as in conventional methods office personnel personally record the data from each and every user as opposed to our system which can handle thousands of concurrent transactions(flow of data) which drastically decreases the time of data collection. It facilitates authority at the higher level to monitor the activities happening in various departments by selecting the fields he/she wants to view. It also provides a platform for managing up-to-date information of various activities being conducted in the Institution and allows every member of the Institution to manage and update his/her information directly in the database.

1.2 Problem Statement

After collection of requirements from faculty and higher authority members of the institutions, we formulated the problem statement. As Albert Einstein quotes “I Would Spend 55 Minutes defining the problem and then five minutes solving it”. The problem statement should be clearly defined before any form of problem-solving can take place. Hence we paid specific attention to formulating the problem statement.

"We want to develop a system that would allow concurrent processing of data entered/removed/modified by users. At the same time maintaining the hierarchy structure of the institution we should allow different parts of data to be entered/viewed/removed by users belonging to a specific hierarchy level in the institution(such as the Principal, the HOD, the department coordinator, the faculty and so on). Further, due to the confidentiality of certain data in the institution, security measures must be undertaken to prevent malicious access to the database." As we proceed further, we discuss measures taken to tackle the problem and get an efficient and practical solution to the same.

1.3 Objectives of the project

As we described the problem statement, we are going to discuss the objectives of our project directly attacking the problem and arriving at a solution.

1. To create a system that can handle concurrent requests.
2. To create a system with minimum downtime and maximum uptime.
3. To create a system that allows for different levels of access to edit/view data pertaining to the hierarchy levels in the institution.
4. To create a system that prevents malicious access to the database.
5. To create a system that generates dynamic reports pertaining to a wide variety of queries.
6. To create a system that easily allows faculty to edit their data.

1.4 Project deliverables

This project delivers a lot of things.

1. Dynamic Report generation: Dynamic report generation is one of the most important functions of the data center. Every year different committees come to college and they expect reports in different formats. Earlier, the faculty had to do the manual work of filling the excel sheets. The datacenter reduces all the manual work which was done by faculty. The query page uses special filters which can get the report with desired information.
2. Get to know your colleagues: A faculty can browse the profiles of there colleagues. They can get to know about their publications and conferences they have published, their expertise.
3. hod dashboard: hod of each department can browse the profiles of each of the faculty and generate dynamic reports.
4. coordinator dashboard: coordinator id is assigned to each department. The coordinator can edit the department information and can generate the dynamic report related to the department. The person using this id can also edit the personal information related to the faculty.
5. admin dashboard: Admin is the superuser who can edit and add any information to any department and generate dynamic reports for any department.

1.5 Current Scope

The current scope is limited to our college. All the faculty members are provided their login ids. They can manage their profile and add research work and publications that they have done. The head of the department can look at faculty profiles and generate dynamic reports for their departments.

The principle is also given a login, where he can browse the profiles of faculty and hods and generate dynamic reports for any department he wants.

1.6 Future Scope

In future, we can extend the data center to support multiple colleges at once. We can add a registration page for college. Using excel importer function import the data of college. This will allow us to support multiple colleges at once online.

The following are the features which we can add to the datacenter.

1. We can use data analysis and find out the most active faculty in a department in publishing conference papers and journals. This will motivate the rest of the faculty to publish research papers and work harder.
2. We can also do data analysis and find out the most active department. This will motivate all the other departments to work harder and increase the skills of students and teachers.
3. The teachers can share their profile link as their resume. We can provide a guest login for the person viewing the profile so that, he can just view the profile of the teacher.

2. PROJECT ORGANIZATION

2.1 Software Process Models

We have used Extreme Programming (XP) as software development model.

Extreme Programming (XP) is an agile software development framework that aims to produce higher quality software and higher quality of life for the development team. XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development.The extreme programming has a lot of advantages. These are the following important advantages that motivated us to use extreme programming

The greatest advantage of Extreme Programming is that this methodology allows software development companies to save costs and time required for project realization. XP eliminates unproductive activities to reduce costs and frustration for everyone involved. It allows developers to focus on coding.One of the major advantages of Extreme Programming is that it reduces the risks related to programming or related to project failure. At the end, XP ensures that the client gets exactly what he wants.
Simplicity is one more advantage of Extreme Programming projects. The developers who prefer to use this methodology create extremely simple code that can be improved at any moment.

The basic advantage of XP is that the whole process is visible and accountable. The developers make concrete commitments about what they will accomplish, show concrete progress. Constant feedback; demonstrate the software early and often, listen carefully and make any changes needed. Sprints help the team to move in the right direction. This approach creates working software faster. Regular testing at the development stage ensures detection of all bugs, and the use of customer-approved validation tests to determine the successful completion of a coding block ensures implementation of only what the customer wants and nothing more.

3. LITERATURE SURVEY

3.1 Introduction

Our project was pretty straightforward. we referred some websites for nodejs and express reference. We also referred some documents for building the datacenter.

3.2 Related Works with the citation of the References

We used Node.js documentation [1] to refer the functionality of Node.js. we also referred express js [2] functionality from documentation provided by express js. Express

js is framework build on top of node js to simplify its use and provide APIs useful for building big applications.

[REFERENCES]

[1] Node.Js Official Website - https://nodejs.org/en/docs/

[2] Express.JS Official Website - https://expressjs.com/en/4x/api.html

4. PROJECT MANAGEMENT PLAN

4.1 Risk Identification

Risk	Probability	Impact	Exposure	Mitigation Steps
There is a risk that the analysis phase of the project is not thoroughly completed and agreed with the faculty and this could result in a partial solution that will require re-work further down the line.	M	M	M	Exert as much pressure on faculty/staff to devote time and effort into all phases of the project.
The difficulty with core systems not being able to provide the data at the right level.	H	H	H	Appoint a person or staff to gather the data and do the manual entry of data into sheets.
Delay in finalizing the analyzing requirements and finalizing entities and attributes for the database.	H	H	H	Analyze the main and necessary entities and start implementing front design and functionality.

Addition of attributes to the existing entities to make sort and search more easily.	M	M	M	Maintain all the SQL functions in a single module that makes it easier to update the database.
Risk of a client/developer not able to provide sufficient time for the project.	M	M	M	Conduct regular team meetings and client meetings to ensure steadiness of the implementation phase.
Risk of application being crashed due to unseen errors in the code or due to code deletion.	M	M	M	Save local copies of properly functioning code repository to minimize loss of code.

Table 4.1.1 Risk Identification

5. SOFTWARE REQUIREMENT SPECIFICATIONS

5.1 Product Overview:

Managing data in a very large and long-established institution is a tedious job as the size of the institution keeps on extending and to maintain the data of every year in a format that is easily accessible becomes very important. RIT Data-Center records all the data regarding the Institution, various committees and departments in the Institution, achievements, and activities conducted by various departments, admission and scholarship details, faculty details etc in a structured manner that can be read, queried and updated in an easier manner.

It provides various levels of authentication and authorization to the members of the Institution to view and update the data in the database. The primary function of the system is to generate dynamic reports for the departments and the Institution in any specified formats for the given requirements whenever required. It facilitates the authority at the higher level to monitor the activities happening in various departments by selecting the fields he/she wants to view. It also provides a platform for managing up-to-date information of various activities being conducted in the Institution and allows every member of the Institution to manage and update his/her information directly in the database.

5.2 External Interface Requirements

5.2.1 User Interfaces

We describe all the user interfaces made available by the system.

The homepage

A home page is shown to the user who visits the application. It provides a clean and easy to use interface. A Login button is provided which is used to login to the application.

Figure 5.2.1.1 Home Page

Login Page

After the user presses the login button it is landed to the login page.Here the user has to provide his/her credentials and login to the system.

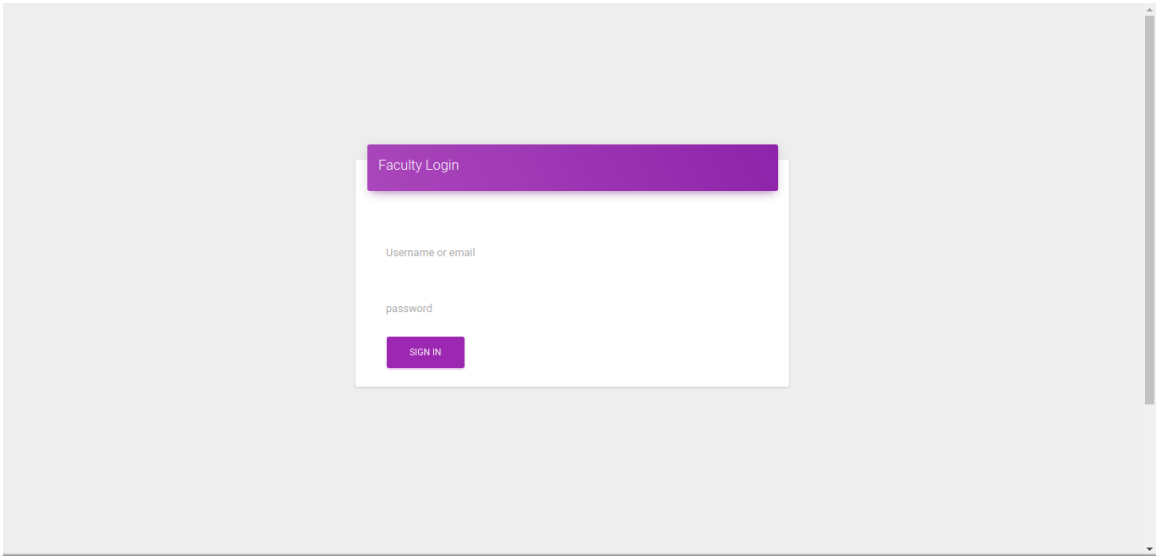

Figure 5.2.1.2 Login Page

Profile Page

Once the user has successfully logged in, he/she is redirected to the user profile page. Here he/she can view or edit the changes with a clean and simple UI.

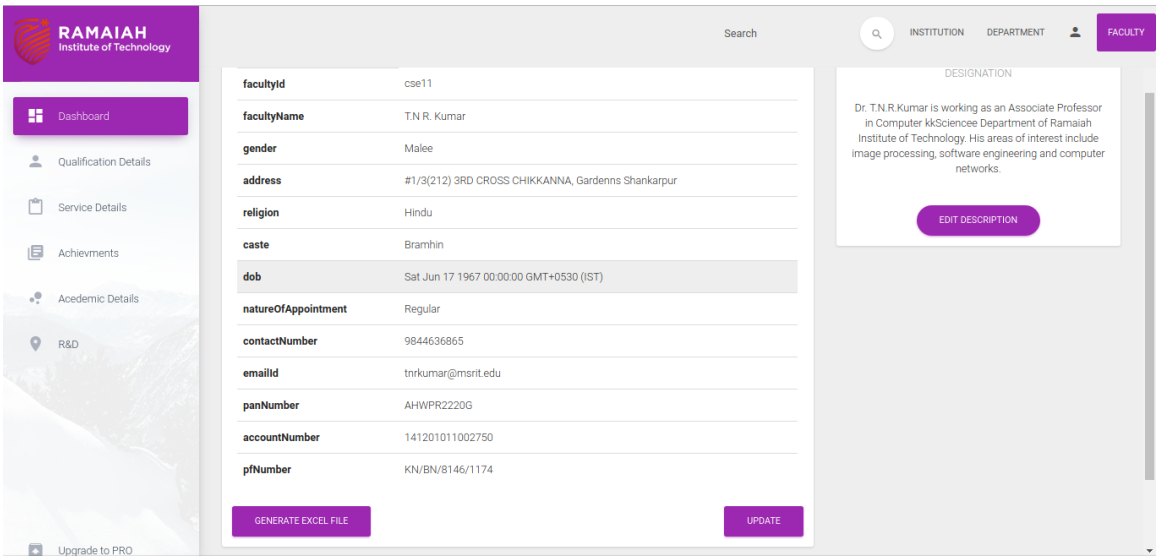

Figure 5.2.1.3 Profile Page

5.2.2 Hardware Interfaces:

Since the web application does not require any designated hardware, it does not have any specific direct hardware interface. The system is based on the client-server architecture. So the connection to the database is handled by the underlying operating system on the web server.

5.2.3 Software Interfaces:

Node JS - for Backend
MySQL - database management
Node JS - For extracting the data from the raw excel sheets
EJS View Engine - For Frontend
HTML/Bootstrap/JS - For making attractive UI

5.2.4 Communication Interfaces:

This website runs on all types of web browsers. Users are authenticated with usernames and passwords.

5.3 Functional Requirements

5.3.1 Dynamic report generation:

The web application should be capable of producing dynamic reports in excel format. In order to obtain such reports, the user will be provided with filters (drop-down menus, checkboxes, etc) which allow him/her to customize the report and download it in excel format.

5.3.2 Authorization levels:

The web application should also provide a secured login to each individual accessing the website. Based on the user's designation and level of authorization, they will be able to manipulate the data contained in different entities of the database. We have categorized the users based on their access rights as:

Faculty - will be having permission to create, modify or delete his/her individual information and can view department level and institution level information. The faculty should not be able to modify or delete information at the department and institution level.

Head of the department - HODs will be given two different logins, one to manage his/her personal information and the other to monitor the department level information. The HOD will also have the permission for generating dynamic reports at the department level.

Department Coordinators / Office Staff - This is the group of users that will be dealing with updating and managing the information at the department level. They will have authority to create, modify and delete data and also generate reports at the department level.

Principal - The principal will be given three different logins, one to manage his/her personal information, second to monitor the department level information and third to monitor the information at the institution level. The Principal will also have the permission for generating dynamic reports at the institution level.

Database Administrator - DBA will have overall access to all the data and will also be responsible for authorizing users.

5.3.3 Extracting data from excel sheets:

The project also covers the crucial part of uploading the humongous amount of data into the database appropriately. This complex task is achieved by a module present in the server, which simply takes in an excel sheet and uploads it to an appropriate table present in the database.

5.3.4 Storing views of recently joined tables:

The principle of caching is used here. Since, there is a high probability of generating reports, which were generated recently, it is a good practice to store the view of the table obtained after joining two or more tables. This decreases the computational time required by the server to join these tables again, hence the user gets his/her reports quickly.

5.2.5 File manager:

There need to be constant checks on the number of excel sheets that are getting stored on the server. For incorporating this mundane task, a module is required on the server side, which keeps a count of the number of sheets present on the server and deletes the files one by one if the count crosses a particular limit based upon the FIFO principle.

6. DESIGN

6.1 Introduction

A design basically showcases how our system will work. It is based on the requirements which we are specifying in our system requirement specifications. In this design chapter, we will outline how the system will implement the features we have specified through the help of some tools such as class diagrams, a sequence diagram and a data flow diagram.

We will also showcase the architectural design of our system along with the graphical user interface which comprises the front end of our project.

6.2 Architecture Design

As we collected the requirements from the clients, there was a strict hierarchy to be followed. Specifically, four levels were required, namely -

1. Super Admin - They got the right to see all the data. They can monitor or query specific details about any faculty/Non-Faculty. They can basically monitor data flowing in and out of the database.
2. Department Coordinator - Department coordinator has the right to view and edit (if needed) any information about faculty in their respective departments. To be noted - they cannot edit other department details.
3. HOD(Head of Department) - They have the right to monitor faculty in their department like any new faculty joining the institution, it will be the responsibility of the HOD to add the following faculty into our database through the system being developed through this project by us.
4. Faculty - This is the final level in the hierarchy system. In this level, the user/faculty will be able to view and will be able to edit their details whenever they wish to edit it. They can insert new details or edit existing ones at their comfort. The system is able to let them use this functionality 24 * 7 all through the year.

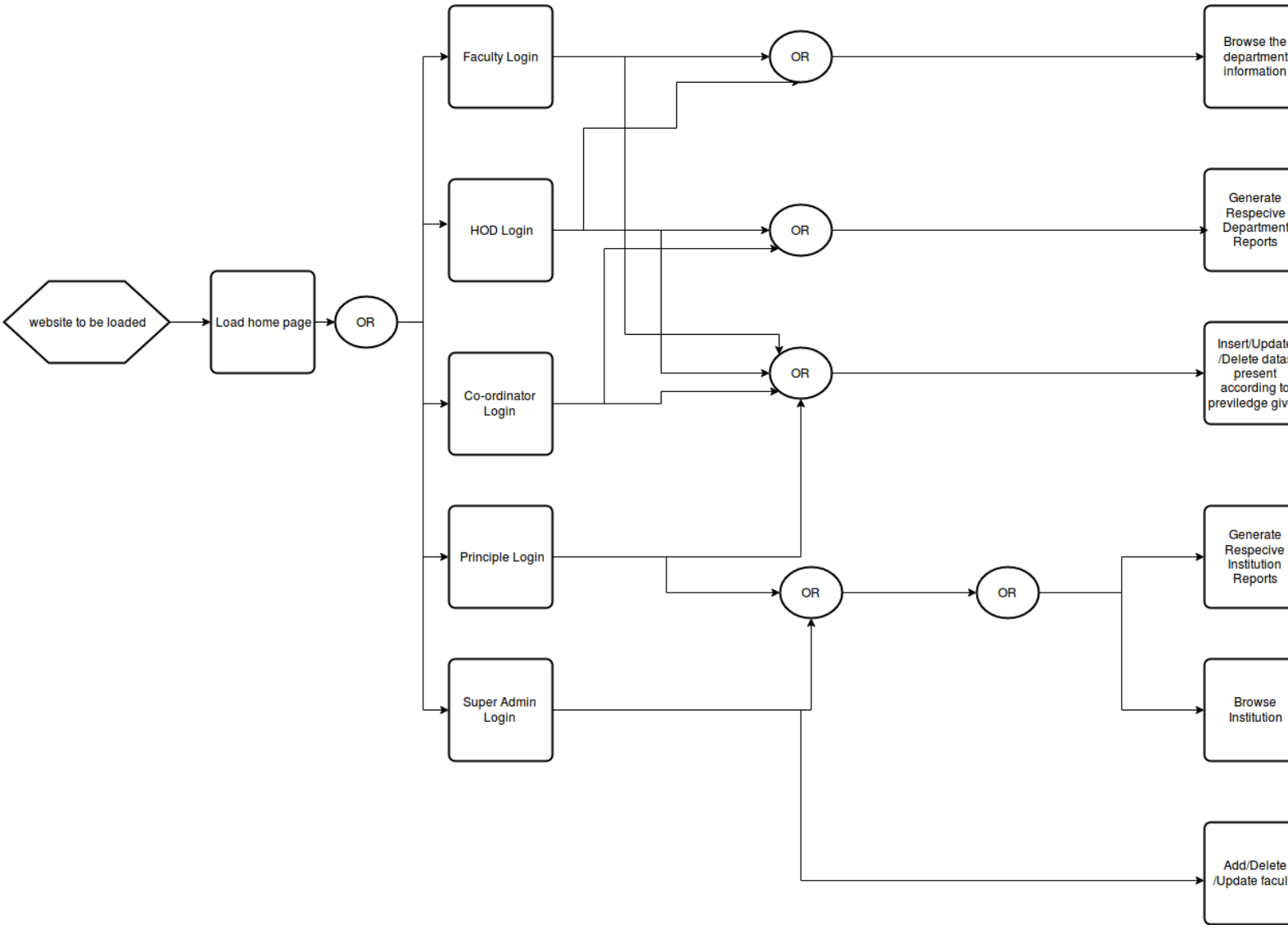

Figure 6.2.1 Architecture Design

6.3 Graphical User Design

This is the login screen. We have integrated all kinds of login ranging from faculty login to the super admin login the same screen for an easy user interface.

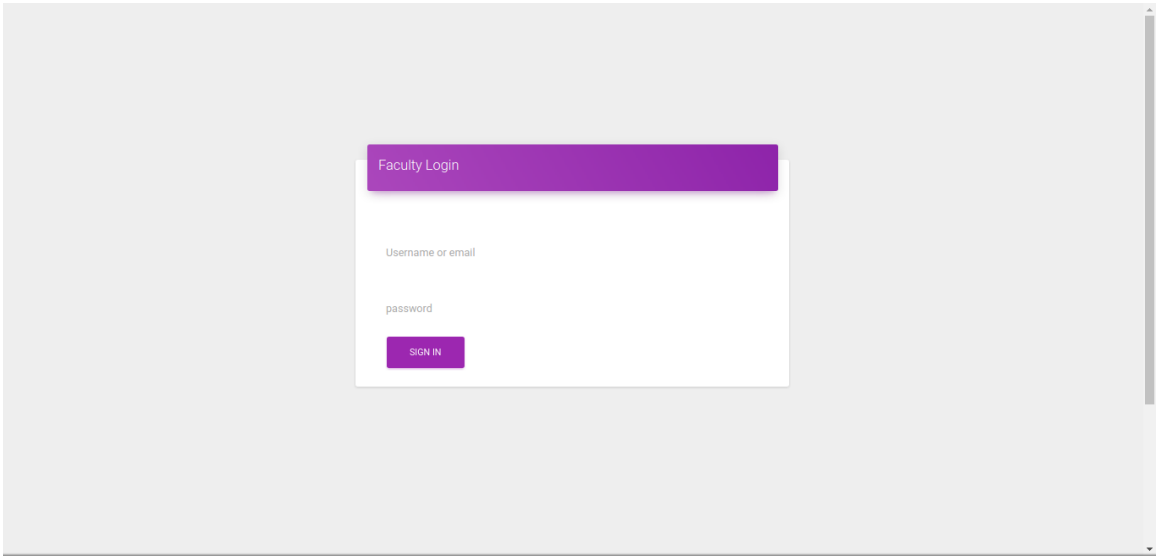

Figure 6.3.1 The Login UI

As portrayed in the snapshots below, we are elegantly displaying the details about you(the faculties) and we also give a beautiful interface to edit any of its details.

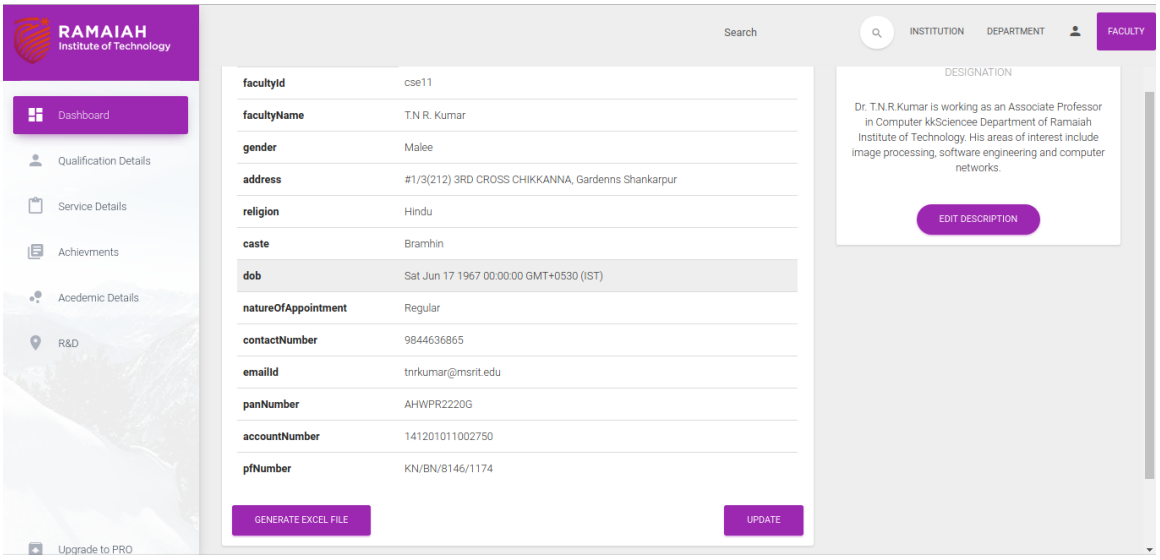

Figure 6.3.2 The profile Page UI

Update your personal details -

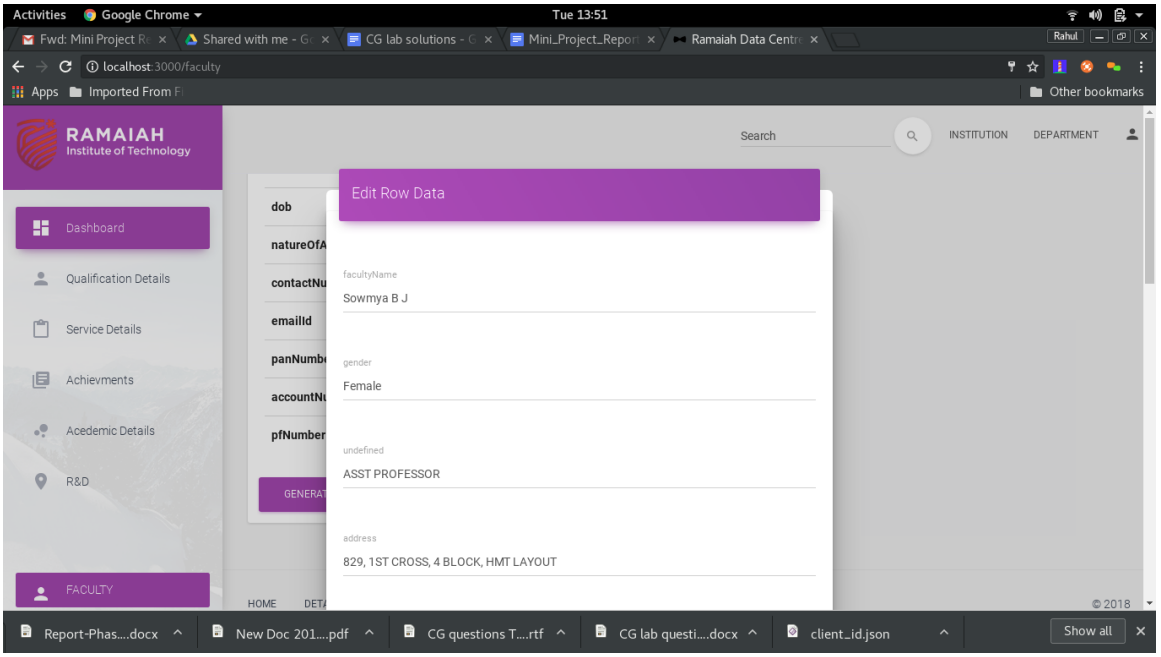

Figure 6.3.3 Update Profile Details UI

Insert new qualifications with just a click.

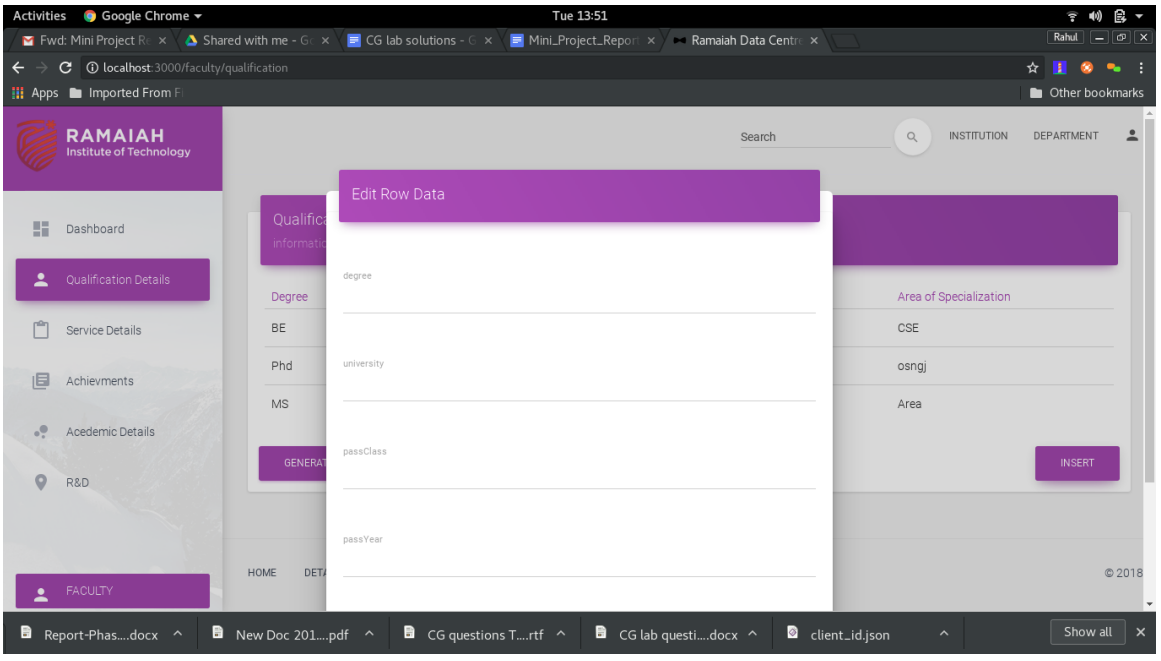

Figure 6.3.4 Insert new details UI

SuperAdmin has now access to the details of the teachers. You can edit update and insert with the super admin privileges.

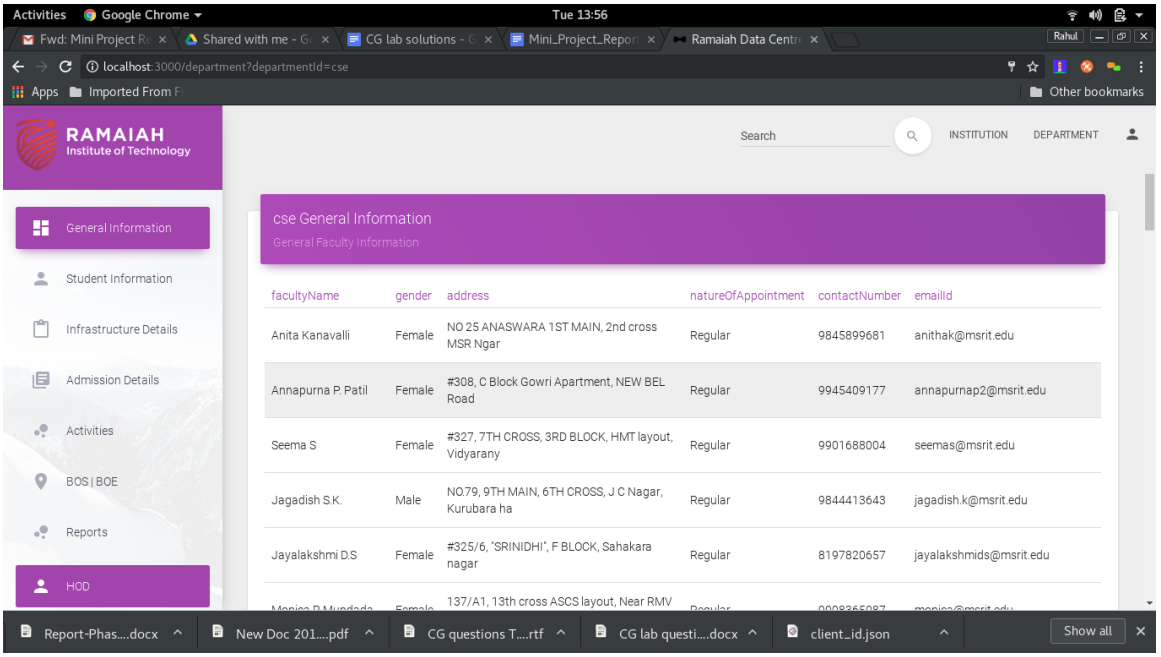

Figure 6.3.5 The HOD UI

Do you have any query? Go for it and see what happens

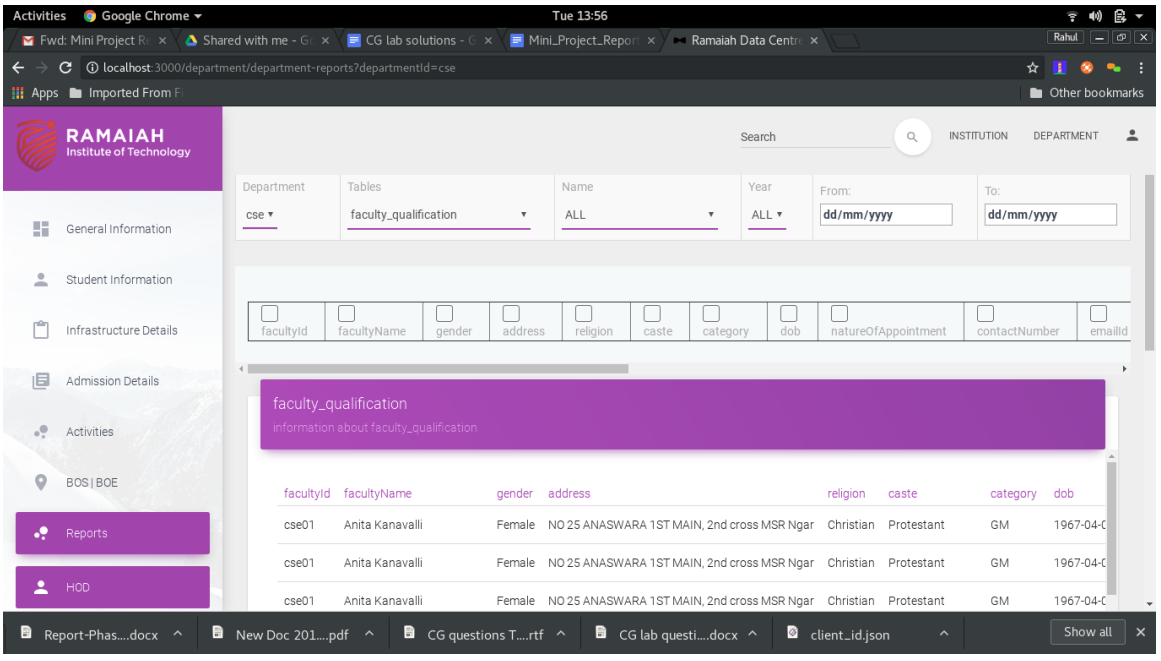

Figure 6.3.6 The Query Page UI

6.4 Sequence Diagram

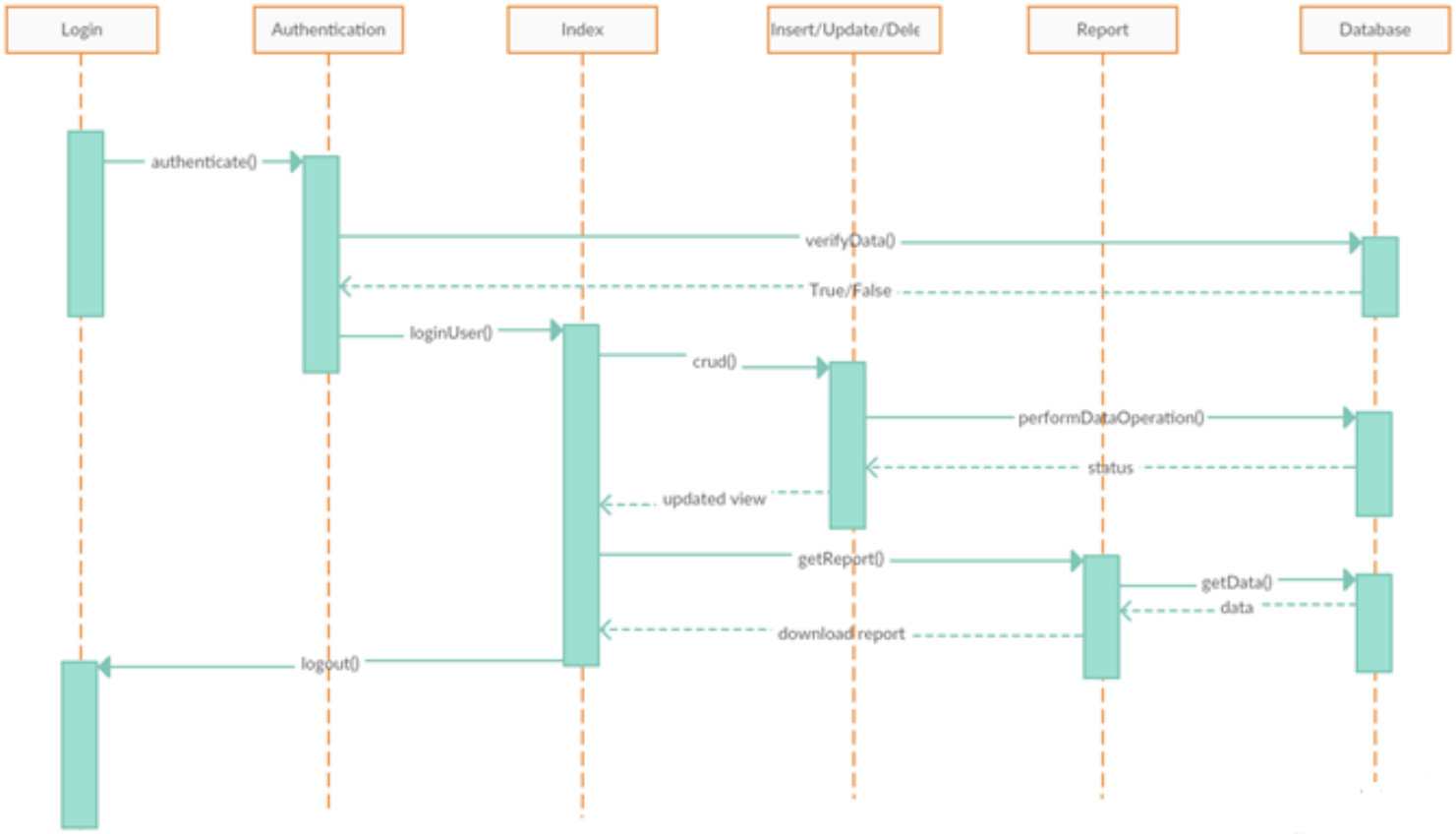

Figure 6.5.1 The Sequence Diagram

6.5 Data Flow Diagram

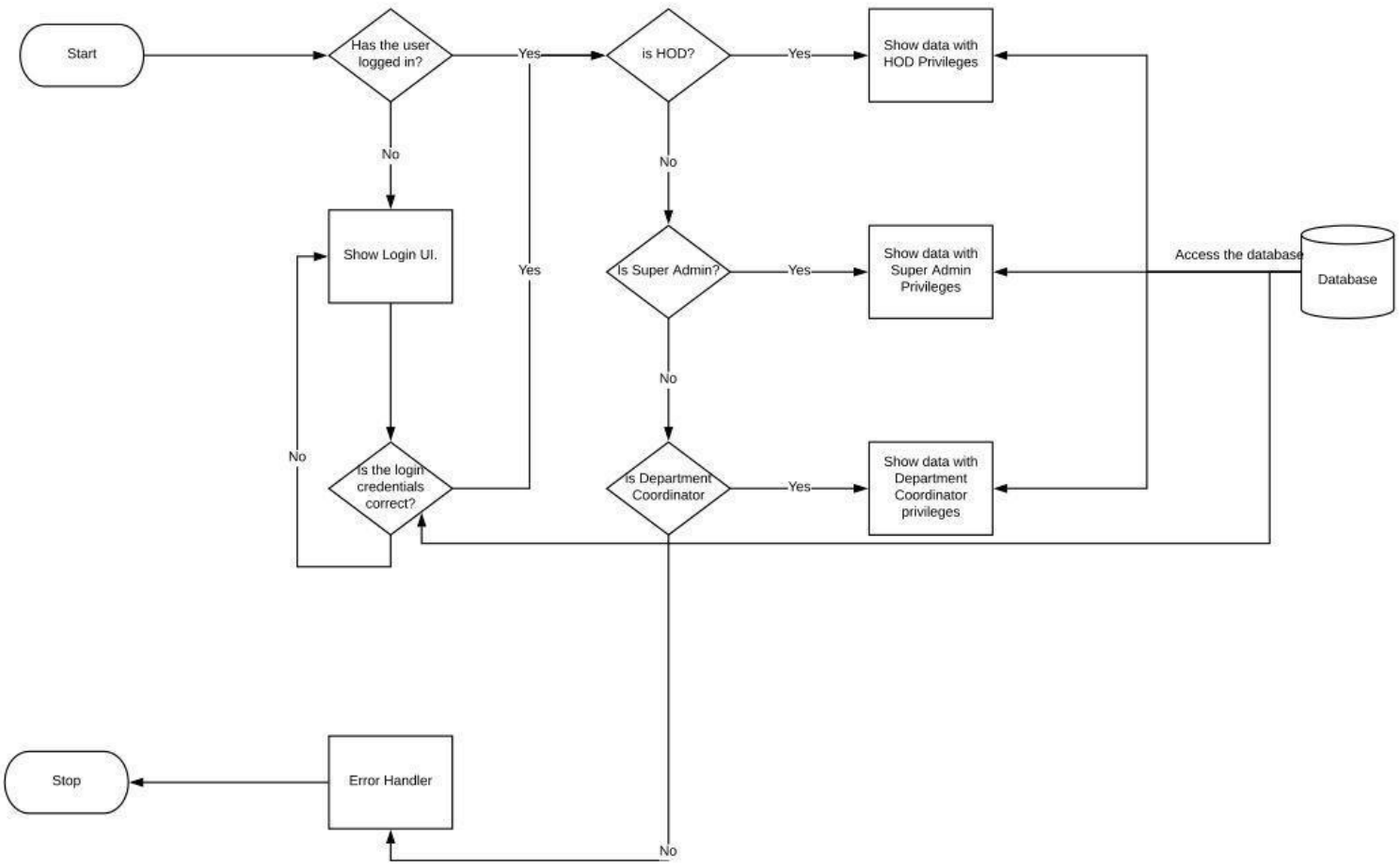

Figure 6.6.1 The Flow Diagram

6.6 Conclusion

We conclude that we have pertained to all the requirements specified by the clients in the srs and developed a robust, practical, solid system which used modern technology as required for the making of a robust, scalable and secure database structure with minimum maintenance time and with a scarce probability of failure.

As per the requirement to deliver a system that can handle multiple transactions at a time and effortlessly processes them without any failure. It has a robust database recovery system employing principles like rollbacking and committing techniques.

7. IMPLEMENTATION

7.1 Tools Introduction

7.1.1 GitHub

GitHub (originally known as Logical Awesome LLC)[3] is a web-based hosting service for version control using git. It is mostly used for computer code. It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.[4]

7.1.2 phpMyAdmin

phpMyAdmin is a free and open source administration tool for MySQL and MariaDB. As a portable web application written primarily in PHP, it has become one of the most popular MySQL administration tools, especially for web hosting services.

7.1.3 Visual Studio Code

Visual Studio Code is a source code editor developed by Microsoft for Windows, Linux, and macOS. It includes support for debugging, embedded Git control, syntax highlighting, intelligent code completion, snippets, and code refactoring. It is also customizable, so users can change the editor's theme, keyboard shortcuts, and preferences. It is free and open-source, although the official download is under a proprietary license. It is based on Electron, a framework which is used to deploy Node.js applications for the desktop running on the Blink layout engine. Although it uses the Electron framework, the software does not use Atom and instead employs the same editor component (codenamed "Monaco") used in Visual Studio Team Services (formerly called Visual Studio Online).

7.1.4 Amazon Elastic Compute Cloud (EC2)

Amazon Elastic Compute Cloud (EC2) forms a central part of Amazon.com's cloud-computing platform, Amazon Web Services (AWS), by allowing users to rent virtual computers on which to run their own computer applications. EC2 encourages scalable deployment of applications by providing a web service through which a user can boot an Amazon Machine Image (AMI) to configure a virtual machine, which Amazon calls an "instance", containing any software desired. A user can create, launch, and terminate server instances as needed, paying by the second for active servers – hence the term "elastic". EC2 provides users with control over the geographical location of instances that allows for latency optimization and high levels of redundancy.

7.1.5 XAMPP

XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac, and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.

7.2 Technology Introduction

7.2.1 Node.js

Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code server-side. Historically, JavaScript was used primarily for client-side scripting, in which scripts written in JavaScript are embedded in a webpage's HTML and run client-side by a JavaScript engine in the user's web browser. Node.js lets developers use JavaScript for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web application development around a single programming language, rather than different languages for server side and client side scripts.

7.2.2 MySQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, Simple Machines Forum, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google (though not for searches), Facebook, Twitter, Flickr, and YouTube.

7.2.3 Express.js

Express.js, or simply Express, is a web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js.

7.2.4 JavaScript

JavaScript, often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm. Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and this is an essential part of web applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it.

7.2.5 Ajax

Ajax is a set of Web development techniques using many Web technologies on the client side to create asynchronous Web applications. With Ajax, Web applications can send and retrieve data from a server asynchronously (in the background) without interfering with the display and behavior of the existing page. By decoupling the data interchange layer from the presentation layer, Ajax allows Web pages, and by extension Web applications, to change content dynamically without the need to reload the entire page. In practice, modern implementations commonly utilize JSON instead of XML due to the advantages of JSON being native to JavaScript.

7.2.6 jQuery

jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. It is free, open-source software using the permissive MIT License. Web analysis indicates that it is the most widely deployed JavaScript library by a large margin.

jQuery syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plugins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, themeable widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and Web applications.

7.2.7 EJS

Easy Java/Javascript Simulations, also known as EJS (or Ejs, or EjsS), is a free authoring tool written in Java that helps non-programmers create interactive simulations in Java or Javascript, mainly for teaching or learning purposes. EJS has been created by Francisco Esquembre and is part of the Open Source Physics project.

7.3 Overall view of the project in terms of implementation

The preliminary step involved in the implementation phase was to decide technology stack we would use to implement the system in this project. We had elaborate brainstorming sessions on deciding the technology stack as it is critical for a large project and it would be catastrophic to change it in the later stages. After rigorous discussion, we finally decided that we will go with the following stack for the implementation of such a large project.

7.3.1 Front End

We decided to use the following technologies and frameworks

1. **Bootstrap:** Bootstrap is a free and open-source front-end library for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only. Bootstrap is the second-most-starred project on GitHub, with more than 123,000 stars The version we used is bootstrap 4.
2. **JQuery**: jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. It is free, open-source software using the permissive MIT License. Web analysis indicates that it is the most widely deployed JavaScript library by a large margin.
   The version used by us for JQuery is 3.3.1
3. **EJS View Engine:** Embedded javascript is a very powerful tool for communicating between the frontend and the backend. This is an integral part of the node and it is the third most use view engine for the node js communicating with the frontend. We are using the version 2.5.7 for our project.

7.3.2 Back End

The backend we used was decided after a lot of discussion and researchers. We need to sit with our fellow faculties for a lot of time to decide what shall be our data structure.

- We had a lot of conflicts in thoughts to whether we need to use the SQL or the NoSQL database.
- But we finally decided to go for the SQL database which is structured database language.
- And Since out project contains structure data it would be good for us to select the database which is relational in nature and hence the structured query language was the optimum choice for the project.

7.4 Explanation of Algorithm and how it is been implemented

Data Cleaning Algorithm:

```
Algo cleanData(inputExcelFile): returns modifiedExcelSheet
    For eachRow in inputExcelFile:
        For eachColumn in eachRow:
            Remove special Characters from entry[eachRow][eachColumn]
            Convert entry[eachRow][eachColumn] to lower case
            //Request teammates to suggest any other changes to clean the data
    modifiedExcelSheet<--entry[maxRow][maxColumn]
    return modifiedExcelSheet
```

7.5 Information about the implementation of Modules

As mentioned in our SRS given by the client, we have implemented all the modules and made sure that our code is highly modular and we don't have any repetition or any redundant code. This is highly required for such big system which we are working as part of the project with our fellow clients.

The modules which we will be discussing will be-

1. Login

This is the module in which we verify the user to login with either of the permissions. The permissions which the users' have is either faculty, or HOD, or the department coordinator, or the super admin pertaining to the hierarchy level described above. It takes care of preventing all the security vulnerabilities. Once a user logs in successfully, it automatically detects the level of the user and redirects the user to the correct dashboard with the correct CRUD permissions the level has.

2. Edit/Update Faculty Details

This module lets the user edit their details. This module also lets the user insert new details like new conference attended, new papers published, new chapters authored, journal publications and so on and so forth. This module also lets the user update existing details. This module takes care of all the user privileges and updates only if the user is privileged to do the action.

3. Delete the details

This module gives the user the option to delete any existing information about them. The module takes care of all the user privileges and deletes only if the user is privileged to do the action.

4. Add new Faculty

This module helps the user to add new faculty members to our platform. This module gives exclusive privilege to the Head of the Department(HOD) and the Super Admin in the hierarchy level.

5. Remove Faculties

This module helps the user to remove existing faculty members from our platform. This module gives exclusive privilege to the Head of the Department(HOD) and the Super Admin in the hierarchy level.

6. Excel Sheet generation

This module helps to export the required data by the user(for example in the query page or at the bottom of all the table view pages) into excel format. This module employs optimization techniques so as to speed up the conversion process. The privilege to use this module is for everyone except the faculty members. In other words, this privilege is given to the Head of the Department (HOD), the Department Coordinators, and the Super Admins in the hierarchy level described in the requirement documents.

7. Parsing the excel sheets

This requirement came into the light because we did not have data in the proper format. Most existing data was in the word document or in the excel sheet format. It was a great challenge for us to manually put such huge amounts of data into the central database. We had to figure out a remedy through which we can tackle this issue. So we decided to make a module to parse the excel sheets of a certain format and put it directly into the database after data cleaning and data preprocessing. This was critical in saving development time by drastically reducing the time taken to populate the central database manually with data.

8. Generating Reports

The trademark feature of our project was to display dynamic reports. Technically, this meant we should allow data to be selected from any table in the database and display only the information requested by the user. It took most of our development time.

7.6 Conclusion

Effortlessly uploading of existing data in raw form (excel form, pdf files, etc) to the central database automatically reduces the time taken to populate the database exponentially. In contrast, conventional methods of manually entering data into the database take huge amounts labor and huge amounts of time. We employed a radical approach to automating the process.

8. TESTING

8.1 Introduction

The appraisal of the software against the requirements which were collected was undertaken. The differences between the existing and the required conditions were noted down and the specific actions were taken. In our project, the testing was done manually and without the help of any automated testing tools. Test cases were prepared based on the modules present in the code and executed. Results were conveyed and any changes required were employed.

8.2 Testing Tools and Environment

The testing tools we used for the project is -

1. Localhost
2. phpMyAdmin
3. PostMan
4. Google Chrome Developer Tools
5. Visual Studio Debugger Tool
6. Amazon AWS console Daemon

8.3 Test cases

We had a lot of test cases in every phase of our development. Starting from the development of each module.

1. Insert/Edit Module - Manual entry of data and checking of the data is inserted into the database.

2. Excel Sheet Generation - We fed our database with some test data and checked if it worked just the way we expected it to work by checking if we have all the rows and columns which we need to be there in the excel sheets, as it's already there in our database.

3. Parsing the excel sheets - For performing this task, we took help of Lab assistant Mr. Arjun from SAP lab of computer science and engineering department of Ramaiah Institute Of Technology. He was given the work to create datasets in the format we asked him to do. The datasets were created in the excel sheet. We fed the data in the excel sheet and saw the result by checking if all the data we require from the excel sheets are inserted into the database properly without any errors.

9. CONCLUSION & SCOPE FOR FUTURE WORK

As we elaborated the problem statement earlier in the report, we conclude that the problem indeed requires a novel solution. We analysed various challenges, and risks involved in different phases of the development of such a system that would solve our problem. We also verified the technology stack used against the requirements of the system. Appreciating the size of the project, we employed a structured model in delivering the product pertaining to the software engineering principles in the current industry. We discussed the detailed schedule and planning done before starting the implementation of the project and we made sure each teammate gets to contribute and work towards getting the system into place. We also discussed various testing methodologies used to test our product, whether it met the requirements of the clients strictly and performed in the desired way in a wide array of scenarios. We also tested how the system recovers from unprecedented errors and also checked the stability of the system under various software and hardware environments.

Also, there is a huge future scope for such a huge project. The project can be extended further to include a host of functionalities that would change the way data is managed in large, well established institutions.

We discuss the future scope of the project.

1. We can use data analysis and find out the most active faculty in a department in publishing conference papers and journals. This will motivate the rest of the faculty to publish research papers and work harder.
2. We can also do data analysis and find out the most active department. This will motivate all the other departments to work harder and increase the skills of students and teachers.
3. The teachers can share their profile link as their resume. We can provide a guest login for the person viewing the profile so that he can just view the profile of the teacher.