

PROGRAMMING 3 FISA Rapport

Prepared by: Israel Mbombole

219120498

Lecturer: Mr. M Adonis

**RETT JOB SEARCH PORTAL**

Jun 2021

**Abstract**

“RETT” is an online Job Search Portal, a web application through which job seekers can register and apply for jobs. Through this portal employer can also post their jobs and review applications. The traditional recruitment systems are time taking and costly. A job seeker must find jobs through advertisements, college fairs, job fairs etc., and the employers must put in much effort to find the right candidate for a vacant position. This application addresses such shortcomings and is a convenient platform for both job seekers to find and apply for jobs and for employers to post jobs and review applications with much ease. Candidates can search for jobs in any field through advanced search capabilities. They can upload their resumes to this application which is stored for future use also. Employers can download these resumes and post/delete job positions. The admin controls this portal and makes the decision about companies and jobs that can access/appear in this portal. Candidates and Employers can use this portal without any geographical barrier, from any part of the world. This application is also developed by using some cutting-edge technologies that are in great demand in the IT industry today.

**Table of Contents**

[List of Figures 6](#_Toc75039953)

[List of Tables 8](#_Toc75039954)

[Acknowledgements 9](#_Toc75039955)

[Chapter 1 - Introduction 1](#_Toc75039956)

[1.1 Project Description 1](#_Toc75039957)

[1.2 Motivation 2](#_Toc75039958)

[1.3 End users 3](#_Toc75039959)

[Chapter 2 - Background and Technologies Used 5](#_Toc75039960)

[2.1 MySQL 5](#_Toc75039961)

[Chapter 3 - Related Work 6](#_Toc75039962)

[3.1 Existing System 6](#_Toc75039963)

[3.2 Proposed System 7](#_Toc75039964)

[Chapter 4 - Requirement Analysis 8](#_Toc75039965)

[4.1 Requirement Gathering 8](#_Toc75039966)

[4.1.1 Functional Requirements 8](#_Toc75039967)

[4.1.2 Non-Functional Requirements 9](#_Toc75039968)

[4.2 Requirement Specifications 10](#_Toc75039969)

[4.2.1 Software requirements: 10](#_Toc75039970)

[4.2.2 Hardware Requirements 10](#_Toc75039971)

[Chapter 5 - System Design 11](#_Toc75039972)

[5.1 Use Case Diagram 11](#_Toc75039973)

[5.1.1: Description of Use Case Diagram 12](#_Toc75039974)

[5.2 Activity Diagram 13](#_Toc75039975)

[5.2.1: Description of Activity Diagram 14](#_Toc75039976)

[5.3 Sequence Diagram 15](#_Toc75039977)

[5.3.1: Description of Sequence Diagram 16](#_Toc75039978)

[Chapter 6 - Database Design 19](#_Toc75039979)

[6.1 ER Diagram 22](#_Toc75039980)

[Chapter 7 - Testing 24](#_Toc75039981)

[7.1 Testing Levels 25](#_Toc75039982)

[7.2 Tests performed on Job-Seeker Module 26](#_Toc75039983)

[7.3 Tests Performed on Admin Module 28](#_Toc75039984)

[7.4 Tests Performed on Company Module 29](#_Toc75039985)

[7.5 Bugs Encountered 30](#_Toc75039986)

[7.6 Load Testing 31](#_Toc75039987)

[7.6.1 Hardware Configuration Used for Testing 32](#_Toc75039988)

[7.6.2 Software Configuration Used 32](#_Toc75039989)

[Chapter 8 - Implementation(GUI) 34](#_Toc75039990)

[8.1 Jobseeker Module 34](#_Toc75039991)

[8.2 Admin Module 41](#_Toc75039992)

[8.3 Company Module 45](#_Toc75039993)

[Chapter 9 - Conclusion 51](#_Toc75039994)

[Bibliography 53](#_Toc75039995)

# List of Figures

[Figure 2.1 Node JS Architecture [1] 6](#_bookmark10)

[Figure 2.2 NodeJS Components [3] 8](#_bookmark14)

[Figure 2.3 AngularJS Architecture [4] 10](#_bookmark18)

[Figure 2.4 Angular Built-in Form Validation 13](#_bookmark21)

[Figure 2.5 Angular Custom Validation 14](#_bookmark22)

[Figure 2.6 AngularJS MVC Architecture [5] 15](#_bookmark24)

[Figure 5.1 Use Case Diagram of Job Search Portal 24](#_bookmark40)

[Figure 5.2 Activity Diagram of Job Search Portal 26](#_bookmark43)

[Figure 5.3 Sequence Diagram of Job Search Portal[6] 27](#_bookmark46)

[Figure 6.1 Company Details Table 31](#_bookmark49)

[Figure 6.2 Job Details Table 32](#_bookmark50)

[Figure 6.3 User Details Table 33](#_bookmark51)

[Figure 6.4 User Job Table 33](#_bookmark52)

[Figure 6.5 Entity Relationship Diagram 34](#_bookmark54)

[Figure 7.1 Load Testing for Job Search Portal on the same system 45](#_bookmark67)

[Figure 7.2 Load test Result for job Search portal with client and server on different system 46](#_bookmark68)

[Figure 7.3 Load test Result for Job Search Portal on the same system 46](#_bookmark69)

[Figure 7.4 Load test Result for job Search portal with client and server on different system 47](#_bookmark70)

[Figure 7.5 Load test Result for Job Search Portal on the same system 47](#_bookmark71)

[Figure 7.6 Load test Result for Job Search Portal on the same system 48](#_bookmark72)

[Figure 7.7 Load test Result for job Search portal with client and server on different system 48](#_bookmark73)

[Figure 7.8 Load test Result for job Search portal with client and server on different system 49](#_bookmark74)

[Figure 8.1 Jobseeker Login Page 50](#_bookmark77)

[Figure 8.2 Jobseeker Wrong Username/Password Error Message 51](#_bookmark78)

[Figure 8.3 Jobseeker Forgot Password Page 51](#_bookmark79)

[Figure 8.4 Jobseeker Homepage 52](#_bookmark80)

[Figure 8.5 Jobseeker Applied jobs 53](#_bookmark81)

[Figure 8.6 Jobseeker Upload CV 53](#_bookmark82)

[Figure 8.7 Jobseeker Upload CV Successful 54](#_bookmark83)

[Figure 8.8 Jobseeker Edit Profile Page 55](#_bookmark84)

[Figure 8.9 Jobseeker Profile Registration Page 56](#_bookmark85)

[Figure 8.10 Jobseeker Profile Registration Error 57](#_bookmark86)

[Figure 8.11 Admin Login Page 57](#_bookmark88)

[Figure 8.12 Admin Home Page 58](#_bookmark89)

[Figure 8.13 Admin View Companies Page 59](#_bookmark90)

[Figure 8.14 Admin Approve/Disapprove Companies Message 59](#_bookmark91)

[Figure 8.15 Admin Job Approvals Page 60](#_bookmark92)

[Figure 8.16 Company Login Page 61](#_bookmark94)

[Figure 8.17 Company Registration Page 61](#_bookmark95)

[Figure 8.18 Company View Jobs Page 62](#_bookmark96)

[Figure 8.19 Company View Applications Page 63](#_bookmark97)

[Figure 8.20 Company Post a Job Page 64](#_bookmark98)

[Figure 8.21 Company Edit Profile Page 65](#_bookmark99)

[Figure 9.1 Lines of Code on the server side 67](#_bookmark103)

[Figure 9.2 Lines of Code on the client side 67](#_bookmark104)

# List of Tables

[Table 7.1 Tests performed on Job-Seeker Module 38](#_bookmark58)

[Table 7.2 Tests performed on Admin Module 41](#_bookmark60)

[Table 7.3 Tests performed on Company Module 42](#_bookmark62)

# Acknowledgements

I would like to thank my lecture, **Mr. Adonis M** for believing in me, my capabilities and His constant guidance, encouragement and valuable feedback led to the successful completion of this evaluation.

And I cannot finish without thanking my Mother and Dad for always believing in me and my abilities, for pushing me a step further whenever I have stumbled. A cheerful thanks to few significant friends without whose constant love, help, support and motivation this task would not have been achieved.

Above all, thank you God for all the opportunities that you have given me.

# Chapter 1 - Introduction

## Project Description

Whether entering the job market for the first time or re-entering after a break or switching career, job search is a challenging task. But how about tools/applications, making this tedious process look friendly, systematic and easy to reach out, to employers or candidates. Searching and landing up with a dream job is a tedious process for a job seeker and on the opposite hand, connecting with desirable candidates best fit for a job position is a challenging and important work for the employers. This project is aimed at making such challenges much easier despite the geographic location of either the job seeker or the Company. Although a job search portal doesn’t guarantee a job offer, it is the best place for potential candidates and employers to get connected and know more about each other. “**RETT**” is an online web application which is a job search portal. It is a simple, efficient, convenient and systematic portal through which job seekers and employers connect with each other. This portal enables candidates looking for jobs to register themselves with the website, look up for different jobs according to their qualifications and apply for those jobs conveniently. Job seekers can also update their details entered during registration as well as their skill sets. On the other hand, employers can register to this portal and publish their jobs which would enable them to find the suitable candidates for their vacant positions. The Company can view the job applications and take necessary steps. Both companies’ registration and company job posting requires Admin approval to be a part of the job search portal. Some of the existing and old-fashioned methods of

recruitment involve advertisements in newspapers, posters, televisions, different job fairs, college career fairs etc. However, such processes are costly and time taking. Handing over paper printed resumes, keeping a track of them, handling and processing them and then getting hold of the desirable candidate to be called for the interview it sounds like a lot of effort and hard work. With the evolution of the world of the Internet and rapid technological advancement, such efforts can be minimized. A job search portal web application comes to rescue at this point where a lot of meaningful time can be saved as well as the cost of advertisements. The entire process of a job searches or a candidate search is speeded up. Manual processes get replaced by automated processes. With job search portals the trend of paper resumes gets replaced by online resumes. These resumes are stored in company databases for future references also. Candidates and employers are just a few clicks away to get connected. Another advantage is once the candidate is registered and applies for a position, his/her information stays with the company database for both the present and future use for available positions. The traditional format of recruitments has been overshadowed with the modern simplistic approaches of e-recruitments.

## Motivation

Visiting company web sites and applying for individual jobs are less motivated and a lot of hard work. I have failed to visit tons of job fairs happening around the country because of time constraints, school semesters etc. Knowing about a company, knowing what kind of qualifications and requirements they want for a position is always so much

time taking. I have always felt the need of friendly applications that gives me all these details in one place and saves a lot of my time. During my undergraduate and some graduate years, the only way I have looked up for jobs is through company websites or employee referrals or through a lot of networking with company personnel. But with the fast rate of technical advancement I have come across many online applications that makes finding a suitable job according to my qualifications much easier, knowing about different positions opened in my desired companies, the qualifications or requirements that the job position needs and search features to retrieve my desired information all bonded in one place. This motivated me to develop an online job search portal as I realized their value as a student and their importance too, as they save a lot of time and effort. Apart from this I was motivated to build this application to learn the usage of some cutting-edge technologies and gain some hands-on experience. I have used, Html, CSS, and MySQL as database and have gained enough experience and exposure working on them.

## End users

**Job Seekers** - First time candidates can sign up or existing candidates can sign in. After signing in, they can search for jobs according to their qualifications, apply to these positions and upload their resumes. They can also update their details at any point of time. **Companies -** Company can register itself with the portal and once approved by admin, can sign in to the portal. They can see if candidates have applied for their job positions posted. They can also post new jobs.

**Admin -** The main person maintaining the website is the admin. He takes the decision to allow a company to register with his website or not and approves/disapproves jobs posted.

# Chapter 2 - Background and Technologies Used

This is a Representational State Transfer API following the Model View Controller architecture, Html5, CSS3. All validations are done with Jsp. I have used MySQL as the database language and MySQL Workbench as the database tool. With an intention to get acquainted with latest cutting-edge technologies this application got that intention reflected in every technical aspect of it. Learning right from asynchronous I/O bound data interaction to dependency Injections in AngularJS this project had been quite a valuable learning process.

## MySQL

MySQL is an open source Relational Database Management System in which all the data are stored in the form of tables. Each table is connected to some other table i.e. has a relation with another table and this relationship is established through integrity constraints. These tables have columns which represents the attributes of an entity and there are rows of data for each column. This is called the database and is connected to the frontend or user interface with the help of controller. For this project I have normalized database tables named user, company, user\_job, job. We can manipulate the data in the tables such as update, delete, alter, modify etc. This is a fast and highly scalable database management System.

# Chapter 3 - Related Work

## Existing System

The existing systems enables jobseekers to search through print media like poster advertisements, newspapers and visual media like television or company websites for employment opportunities. This is a tedious task as it takes a lot of time and energy to search for the right job position, learn about the position and about the company. Job search for proper match of skill set and salary is challenging. Job seekers can also find jobs through job fairs where they must first make it possible to attend the fairs which might be sometimes impossible with their schedules and if they visit the fairs they must hand over paper printed resumes. The more the number of candidates the more the number of papers for the company which is a lot of manual effort. Again, jobseekers might get job offers through placement cells in respective colleges but getting hold of the right opportunity at the right time is always challenging. On the other hand, the same goes for employers who are looking for candidates who are best fitted for their job positions. They must constantly advertise, go to a lot of job fairs which still doesn’t guarantee the best way to select from a large pool of candidates. Such conventional and outdated systems are replaced by several well featured national job search portals like Monster, Dice.com, Glassdoor, Indeed etc. All these job search and advertisement portals aims at e-recruitment by providing several simple and useful features to jobseekers and employers making job search and candidate selection a much time saving and easier process.

## Proposed System

With the advancement of technology job seekers are relying greatly on Online Job Search Portals. Taking motivation from the conventional systems and their drawbacks and inspiration from the existing job search portals, I decided to develop “Dreams Job”. In the proposed system we are trying to develop an online job search web application that reduces challenges for job seekers to find a desired and suitable job according to their qualification. We aim at reducing the challenges by providing advanced search features that gives the candidate ample scope to select jobs that matches their skill set and requirements and gives them back the exact jobs that are available. This in turn is less time taking as the candidate gets all details in one place and do not have to go to company website to learn about the positions. In the proposed system job seekers can upload their resumes in the required file format, see all the available jobs and search for desired jobs and then apply for those jobs. On the other hand, this system enables employers to post their jobs and get a list of all applications which they can screen online and that reduces the huge amount of manual effort and time. Online recruitment or e-recruitment is turning out to be both the job seekers and the employers’ favorite activity as offer and demand are well met at one place and both must spend less time to get hold of the right roles or candidates. The company can post jobs, see applications and check resumes in the proposed system.

# Chapter 4 - Requirement Analysis

## Requirement Gathering

In the software development life cycle (SDLC), Requirement analysis is the first step of major importance. The entire concentration is on gathering the functional and the nonfunctional requirements for the product to be developed and estimating the feasibility of those attributes. Through requirement gathering we ensure that we are setting project goals and objectives much earlier. Complete understanding of the requirements leads to the successful development of the software. If we don’t do this step, then however hard we work we will never arrive at the desired final product. This is most crucial as without knowing the exact requirements the final output can never be achieved as desired. For this project, I did a major research on the existing system and discussed the functionality that I wanted to develop with my major professor and finally concluded on a concrete set of requirements that I wanted to see as an outcome of my project.

### Functional Requirements

The proposed system has the following functionalities:

* Job Seeker Sign In
* Job Seeker Sign Up
* Job Seeker can view all available jobs
* Job seeker can upload resume
* Job seeker can apply to more than one jobs.
* Job seeker can view all applied jobs.
* Job seeker can update their profile information.
* Admin approves the company registration,
* Companies can post jobs.
* Companies can see applications.
* Companies can see all approved jobs.
* Companies can download resumes uploaded.
* Admin can view all jobs.
* Admin approves the jobs posted by the company.

### Non-Functional Requirements

The major difference between Functional and Non-functional difference lies in two keywords. “What” and “How”. While what the system does is described by its functional requirements, how the system achieves those quality attributes are discussed by the non- functional requirements. The non-functional requirements focus on the scalability of the system, security of the system, reliability and maintenance of the system which are ensured by verifying and validating the system. The proposed system meets the non-functional requirements like security by several form validations and password encryptions, reliability by maintaining integrity with error messages, controlling access of the users. The application is made more scalable with an advanced search feature that filters and delivers

requested data from a huge amount of data. Since the application code is modular in nature maintainability and reusability is also ensured.

## Requirement Specifications

Requirement Specification is an important step in the software development life cycle. Although various software and hardware specifications can be used to develop this application, the software and hardware specifications that I have used to develop this job search portal are mentioned below:

### Software requirements:

* Operating System: Windows 10
* IDE/Text Editor: Sublime Text, Atom, Git Bash CLI
* Application Server: Node 3.10.10
* Frameworks/APIs: AngularJS, Sequelize ORM, Swagger, MySQL Workbench
* Database: MySQL
* Front End: HTML5, CSS3, Java
* Web Service: RESTful web services
* Browser: Preferable Google Chrome or Mozilla Firefox

### Hardware Requirements

* Processor: Intel core i5
* Processor Speed: 2.50 GHz
* RAM: 8 GB
  1. **Sketches** or **prototypes**

Login form

Email

Password

Register form

First Name

Last Name

Email

Password

Work experience form

Work Experience

Reference

Qualification form

Field studied

Period

Institution

Place

Level

Qualification

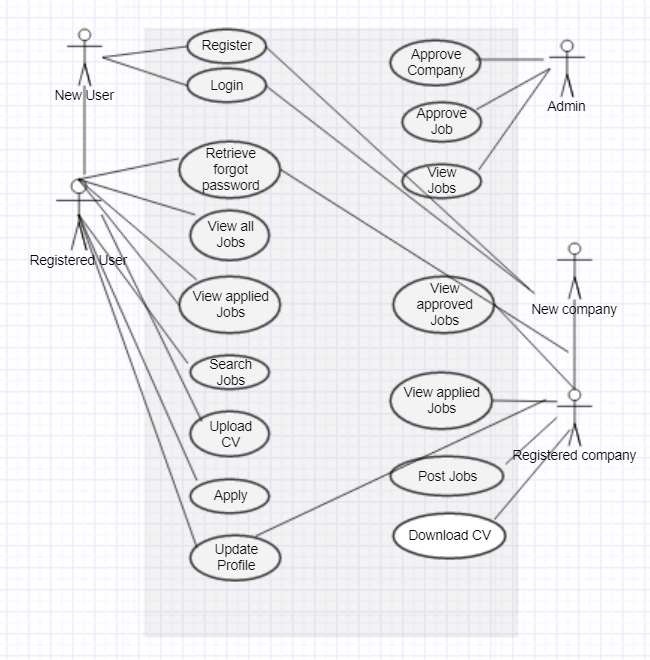
# Chapter 5 - System Design

In the software Development Life Cycle, the output of the system requirement analysis phase can be considered as an input to the system design phase. The architectural description of a System with details about its components and sub components is called System design. The pictorial description of the system, modules, and sub systems gives a proper overview of the system and its design details. Through UML diagrams we specify and visualize various aspects of a System and its architecture. Security, Reliability, being able to deliver desired output to end users based on available resources and that the system is responsive and dynamically changes are some of the important aspects ensured by the

System Design.

## Use Case Diagram

While understanding only the static nature of a system is insufficient, Use-Case diagrams helps to give the dynamic view of the system. Use Case diagrams models the system and the subsystems of an application. There are some external and internal factors that marks the dynamic nature of the Use Case diagram. We call them actors. While Use case diagrams can be considered as a high-level requirement analysis of the system, they give a clear notion of the actors and their roles (use cases) and hence is an important pictorial representation to understand system specifications early in the project. Use case diagrams are a clear visualization of actors (the internal or external factors), their roles (use cases) and relationship amongst these actors and their roles.



**Figure 5.1 Use Case Diagram of Job Search Portal**

### 5.1.1: Description of Use Case Diagram

In this system there are 5 actors namely index, welcome, register, qualification, work experience. The different use cases are:

**Index**- once the user is on the page he or she must register first and get access to login using their credentials.

**Welcome-** once registered they will be taken to a page called welcome

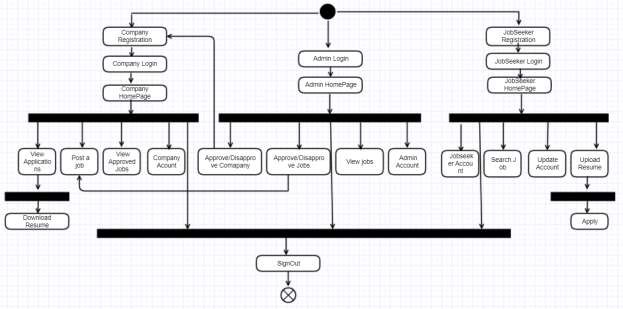
**Register:** the user register using his first name, last name, email and password.

**Qualification**-the user after sign in he or she can now complete the form where he or she specifies her/his qualification

**Work experience**- the use will have to lastly fill his/her work experience in a form call work experience.

## Activity Diagram

Activity Diagram is also one important UML diagram that gives the flow of execution of the system. While not being exact flowcharts activity diagrams have some capabilities like branching or swim lanes or indicating parallel flows. It is a pictorial representation of the different activities of a system, giving the wholistic view. A concept of forking and joining is used inside the activity diagrams to show the activity of the different components of the system. A function performed by the system can be called an activity of the system. Once we make out a mental layout of the entire flow, we proceed in drawing the activity Diagram.



**Figure 5.2 Activity Diagram of Job Search Portal**

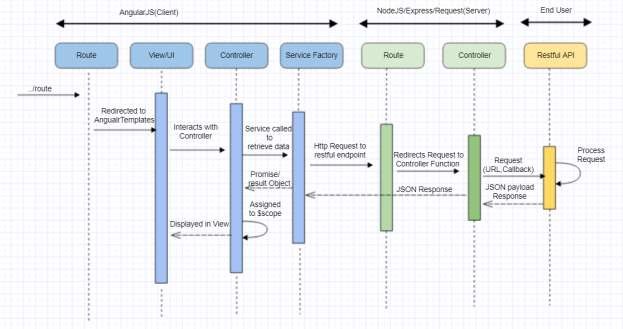
### 5.2.1: Description of Activity Diagram

In the above Diagram it is clearly seen that there are three flows, one of the admin, one of the jobseeker and one of the Company. Once the admin logs in and views his homepage he can approve/disapprove company’s registration, he can approve/disapprove certain jobs from being posted, he can view the job listing and overall, he can dwell in his own account. Once the student wants to login, he first should register himself/herself to the portal, then reach his homepage. There the user can view all jobs, make an advanced search for the desired jobs, upload his resume to the portal and apply for more than one jobs. User can also edit his profile information once he is in his account. Overall, user can dwell in his own profile. Again, if the company wants to be on the portal he first registers himself with the portal and waits for admin approval. Once approved the employer can login to his

homepage, post jobs, view all applications, download resumes and update its profile information. Overall, the employer can dwell in his own account.

## Sequence Diagram

The sequential flow of a system along with its sub system is pictorially represented by the sequence diagram. As the following diagram is an overall system sequence diagram, sequence diagrams can also be drawn at the modular level for every component in the system. Sequence diagrams emphasize more on the system requirements than on the system design. It focuses more on the sequence of messages delivered just after a sequence of activity occurs. Overall a sequence diagram helps in modelling and documenting how a system should behave and helps in validating the logical behavior of complex operations and functions.



**Figure 5.3 Sequence Diagram of Job Search Portal[6]**

### 5.3.1: Description of Sequence Diagram

This sequence diagram describes sequences/actions amongst the front end, backend and an end user.

There are altogether 8 view templates used for this system. The view templates are:

**Index.jsp** -interacts with the user and login

**Register.jsp** -capture Applicants details

**Welcome.jsp** -interacts with the user after registration

**Qualification.jsp** -interacts with the user and capture qualification details

**View.jsp** -interacts with the use and job Application

**Work.jsp** –capture work experiences

**Apply.jsp** –capture job Application details

**Logout.jsp**- interact with the user to logout

**Company Controller Routes:**

* *saveCompany*
* *authenticateCompany*
* *updateCompany*
* *fetchCompanyById*
* *fetchJobSeekersbyCompanyId*
* *fetchCompanyByUserId*
* *VerifyUserorComapny*

**Job Controller Routes:**

* *SaveJob*
* *fetchJobsByStatusAndCompany*
* *dropJob*
* *fetchJob*
* *fetchJobforSeeker*
* *fetchAppliedjobsforSeeker*

**User Controller Routes:**

* *SaveUser*
* *AuthenticateUser*
* *FetchUserById*
* *UpdateUser*
* *FetchUserByEmailId*
* *FetchCustomSearchJobs*
* *updateUserAsCvUploaded*
* *verifyUserorCompany*

# Chapter 6 - Database Design

A good database design is the backbone of a good web application. The “good” is defined by some attributes like a normalized database, integrity constraints well defined, well defined relationships among different database tables. When one database table is related to another database table or to more than one, we call it a relational database system. When we define the term “normalized” we mean that there are no data redundancies i.e. repeated data. We can also define the term “constraints” like some attribute can be null, while another can never be null. Also, sometimes one attribute can have unique value while another may not. There are identification columns in each table which are unique, and the table is represented by that attribute. We call such attributes primary keys. While there are others table which also contains the same primary key for reference and we call them foreign keys. I have used MySQL Workbench as the database tool and MySQL as the database language. I have created 5 tables in the database which I have identified to make the application work well namely:

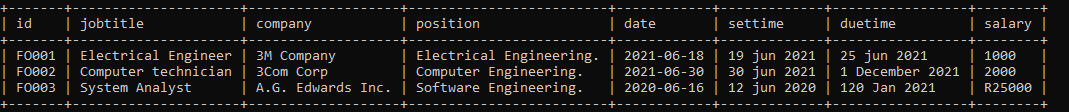
**Company**: It stores details about the company that registers itself to the portal and is approved by the admin/disapproved by the admin by a special flag column called status.



**Figure 6.1 Company Details Table**

Here companyId is the identification number for each company and it is unique. This is the primary key. Next, we have the companyName, email id for the company to access the portal, website of the company, mobileNo which are self-explanatory. We have a column called status which is either 0 or 1.1 means that the company has been approved by the admin and 0 means that the company has been disapproved by the admin. The other field is called the password. Here password has been encrypted by AES encryption strategy so build a secure and robust database, such that at no point of time any user can get hold of the password except the admin and there is never a security breach.

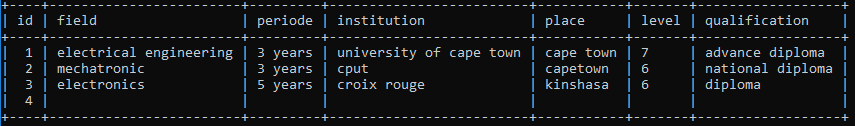
**Job**: This table contains details about the jobs that are posted by the company and both approved/disapproved by the admin by a special flag column called status.



**Figure 6.2 Job Details Table**

This table has a job\_id column which is the unique key for each job and is the primary key for the table. The other columns are title, location, experience required for the job, type of contract i.e. permanent/contract/intern identified as 0,1,2 qualification required for the position, skills required, job posting date, job joining date, contact number for job related queries, salary for the job, company Id which establishes the relation as to which company posted which job and the status which shows whether the job is approved by the admin or not.

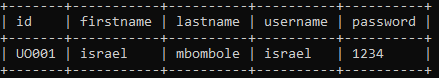
**User**: This table contains user details who registers and logins to the portal.



**Figure 6.3 User Details Table**

There is userId which is identified as the primary key for this table. The other columns include username, email to login to the portal, mobileNo of the user, qualification details of the user, password to login, which is again encrypted to adhere to security benefits related to the database and a isCVUploaded column represented as 0/1 without which user cannot apply for any job.

**User\_job :** This table represents the jobs to which the user have applied.

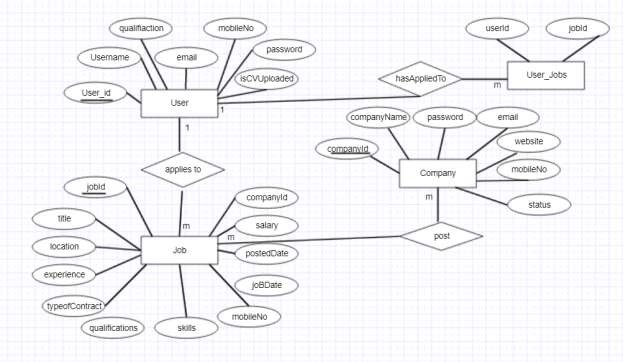


**Figure 6.4 User\_Job Table**

This table has two columns namely userId and JobId which acts as the foreign key for the table. The primary key reference dwells in the User table and Job table respectively. This table helps us to map the user to his/her applied jobs i.e. represents job applications of the user.

## 6.1 ER Diagram

The entity relationship diagram gives a pictorial representation of all the database tables and the relation between the entities. It also shows the cardinality i.e. the many to one or one to one or one to many relationships amongst the tables. This is the first step in designing a database. All the idea of requirements and specification details about the different entities in the database are conceived in the beginning and then transformed into a diagram. This step takes time but once finalized, a good, strong and robust database is a cake walk.



**Figure 6.5 Entity Relationship Diagram**

This diagram represents the relationship amongst all the entities in the job portal. The User applies to jobs and the company posts jobs and the user has applied to certain

jobs. The tables are represented in the form of rectangular boxes as shown in the diagram. The primary keys are underlined. The relationships are shown through diamond boxes. The cardinalities are mentioned through the numbers.

# Chapter 7 - Testing

In the software development life cycle, after the requirement analysis, feasibility study, design and coding phase, one of the phase of utmost importance is the testing phase. In this phase, we get to see if the expected and final outcomes are same with all the required specifications being maintained as per the requirement. This is the debugging phase of the application on its entirety and not simple debugging of few lines of code on the IDE. This phase starts with the smallest unit of a code through Unit Testing and travels a long way till it ends with User Acceptance Testing. For industrial applications, testing is performed through highly automated tools like Docker for Java applications, but such automated testing tools are out of scope for this project. This testing phase helps in finding bugs which can be at the code level, system level, environment level, then fixing those bugs, then retesting the same functionalities to check the new changes are compatible with others and then at the end of many testing strategies the product is delegated to corresponding authority for release. Testing at the application/code level is performed by writing test cases. To err is human – we all know. When code is nothing but human typed language with a sense of grammar and context understandable by the OS and computer scientists, there can always exist errors. We can eliminate these errors by thinking about corner cases or scenarios that might never happen but if happens, our application is strong enough to handle them. Testing the application with every possible scenario which it can/cannot

handle and still our application staying steadfast is what we target for.

## Testing Levels

**Unit Testing –** Testing an application to its smallest unit is called Unit Testing. Again, testing each module of an application which numerous test cases and checking validations against unforeseen scenarios is what unit testing is all about. Once a bug is detected, that is recorded in the bug tracker, a ticket is raised, this bug is fixed, and again new unit test cases are written to perform unit testing over the debugged piece of code.

**Integration Testing** – Once each individual part of the system is tested, every smallest unit is tested, different modules of the system are now integrated together and tested. Whether the integration works or whether a part of the system that is functional individually starts failing when integrated with another part is what integration testing is all about.

**System Testing** – That an integrated system meets all its specifications and requirements is decided by system Testing.

**Regression Testing** – Once the system is debugged, it is tested again to see if it is compatible with the changes made and compatible with any changes made to the environment.

**Load Testing** – Testing that the system can take as much load as it is supposed to take and testing how much load it can take and to what extent it can exceed its limit and where it breaks.

**Performance Testing** - Testing how the system performs like slow/fast and how it performs under certain workloads

## Black box testing

|  |  |  |  |
| --- | --- | --- | --- |
| **MODULE** | **TEST CASE** | **EXPECTED RESULT** | **OBSERVED RESULT** |
| Job-Seeker | New candidate tries  to login | Username does not  exist | PASS |
| Job-Seeker | Provides wrong  password | Username/password  does not exist | PASS |
| Admin | Provide Username and password | Enters admin homepage with all  companies listed | PASS |
| Admin | Click the view companies button | Results in viewing all the approved companies by the  admin | PASS |
| Company | New candidate tries  to login | Username does not  exist | PASS |
| Company | Provides wrong  password | Username/password  does not exist | PASS |

## White box tests performed on Job-Seeker Module

|  |  |  |  |
| --- | --- | --- | --- |
| **MODULE** | **TEST CASE** | **EXPECTED RESULT** | **OBSERVED RESULT** |
| Job-Seeker | New candidate tries  to login | Username does not  exist | PASS |
| Job-Seeker | Provides wrong  password | Username/password  does not exist | PASS |
| Job-Seeker | Provides invalid  username | Username/password  does not exist | PASS |
| Job-Seeker | Registers to the  portal | Registration  Successful | PASS |
| Job-Seeker | Clicks login button  to see user homepage | User Homepage  with all jobs listed is available | PASS |
| Job-Seeker | Forgets password  action | Receive email with  password | PASS |
| Job-Seeker | Misses filling a field in registration  form | Receive error message to fill the  blank | PASS |

**Table 7.1 White box tests performed on Job-Seeker Module**

|  |  |  |  |
| --- | --- | --- | --- |
| Job-Seeker | Misses filling a field in registration  form | Receive error message to fill the  blank | PASS |

|  |  |  |  |
| --- | --- | --- | --- |
| Job-Seeker | Enter 10-digit  mobile number for registration | Receive error  message if not entered. | PASS |
| Job-Seeker | Click the view applications tab | a window opens with all jobs applied  to. | PASS |
| Job-Seeker | Click the uploadCV tab | modal box appears to upload cv with specified file  extension | PASS |
| Job-Seeker | Click the upload button | Receive confirmation message for CV  upload | PASS |
| Job-Seeker | Click the apply button | Receive confirmation  message | PASS |
| Job-Seeker | Click the update profile to edit information | Receive confirmation message after  editing | PASS |
| Job Seeker | Click the sign-out  button | User is logged out  of the system | PASS |
| Job Seeker | Click on the View  Jobs button | Results in list of all  jobs | PASS |

## White box tests Performed on Admin Module

|  |  |  |  |
| --- | --- | --- | --- |
| **MODULE** | **TEST CASE** | **EXPECTED**  **RESULT** | **OBSERVED**  **RESULT** |
| Admin | Provide Username and password | Enters admin homepage with all  companies listed | PASS |
| Admin | Click the view companies button | Results in viewing all the approved companies by the  admin | PASS |

|  |  |  |  |
| --- | --- | --- | --- |
| Admin | Click the sign-out  button | User is logged out  of the system | PASS |

**Table 7.2 Tests performed on Admin Module**

## White box tests Performed on Company Module

|  |  |  |  |
| --- | --- | --- | --- |
| **MODULE** | **TEST CASE** | **EXPECTED**  **RESULT** | **OBSERVED**  **RESULT** |
| Company | New candidate tries  to login | Username does not  exist | PASS |
| Company | Provides wrong  password | Username/password  does not exist | PASS |
| Company | Provides invalid  username | Username/password  does not exist | PASS |
| Company | Registers to the  portal | Registration  Successful | PASS |
| Company | Registers with  existing credential | User /email already  exists | PASS |

|  |  |  |  |
| --- | --- | --- | --- |
| Company | Click the view applications tab | a window opens  with all jobs applied to | PASS |
| Company | Click the view jobs tab | Shows all the jobs approved and  posted | PASS |
| Company | Click the post a job  tab | A form to post a job  appears | PASS |
| Company | Submit the job | Success message  appears | PASS |
| Company | Click the update profile to edit information | Receive confirmation message after  editing | PASS |
| Company | Clicks sign out | Logged out of the system | PASS |

**Table 7.3 Tests performed on Company Module**

## Bugs Encountered

While I performed initial testing on this application, I have faced few bugs and have fixed them. In the user registration form, initially I have not made the e-mail field unique as a result of which users could register with the same email id again and again. I fixed this bug by making the field unique and this validation helped make the application more robust and secure. While testing the salary range bar in the advanced search feature, a bug appeared where the minimum and the maximum range value was constantly showing 0

despite proper values being given at the implementation. I fixed this bug by adding an extra check for the max and min values and the functionality worked perfect. While I was testing the functionality for posting a job, I encountered a bug that returned undefined whenever I tried posting a job. I fixed this bug by returning a promise/callback function in the code, a scenario that I was not handling earlier. While testing the Resume upload functionality, a bug occurred as we could upload files with any extension. I fixed this bug by limiting the file extensions that are allowed.

I also did some functional enhancements which helped improve my software quality too. Initial testing revealed that with a normal search functionality implemented the search space was not narrowed down. After discussing with my advisor, I implemented an advanced search functionality, that greatly helps in narrowing down the search space by filtering the exact requirements of the user. While the company could post a job, initially I did not implement the ability of the company to delete that Job. I also worked on enhancing the company functionality by implementing a trashing system with which the companies can delete the jobs if they wanted to.

## Load Testing

Estimating and eliminating risk from a software is the primary goal of testing. Load Testing is a kind of performance testing that implies applying normal stress to a web application and expecting it to perform as it is intended to perform. Testing how much a time a system takes to load a page and several other technical statistics gives the user an idea about system functionality and the risks that may/may not be associated with it. Load Testing can only be performed at the end of a system production cycle as the performance

of a system and how much users are engaged can be properly estimated and then tested. Load Testing gives us several information like concurrency level which means how many users can a server handle at the same time i.e. concurrently, mean latency which means the average of the response time of a system minus the processing time of the system, number of completed requests and the requests handled per second as well as the percentile of requests served within a certain time. To test this application, I have used a module called load test. We can perform the load test sending the maximum number of requests and the desirable concurrency level and sending out the server URL that we want to perform load test on. Load test also allows us to set requests per second.

### Hardware Configuration Used for Testing

* Operating System: Windows 10(64 bit)
* Processor: Intel core i5
* Processor Speed: 2.50 GHz
* RAM: 8 GB

### Software Configuration Used

* MySQL version 5.7

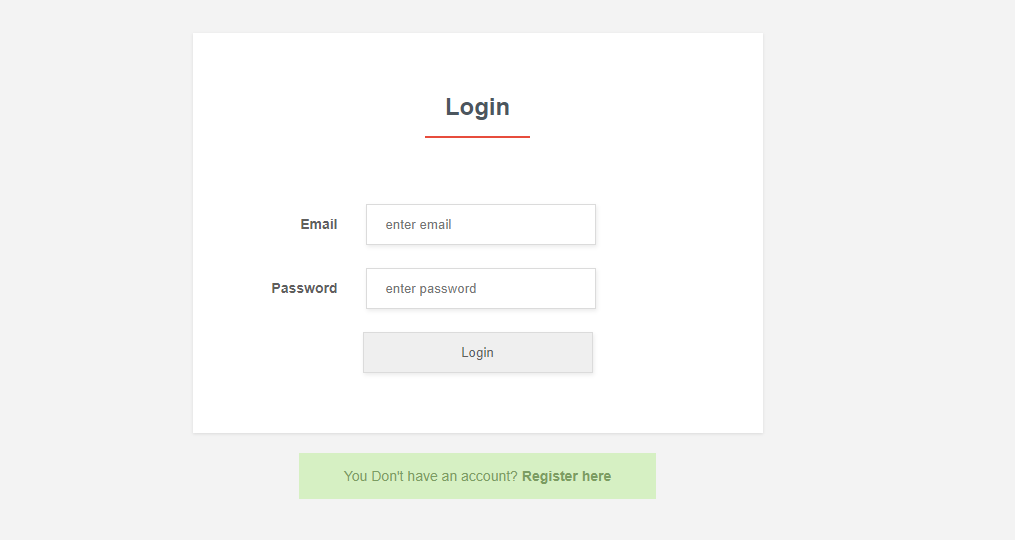
# Chapter 8 - Implementation(GUI)

This job search portal has three basic modules namely User, Admin and Company. A sequence of screenshots for the graphical user interface gives a clear representation of

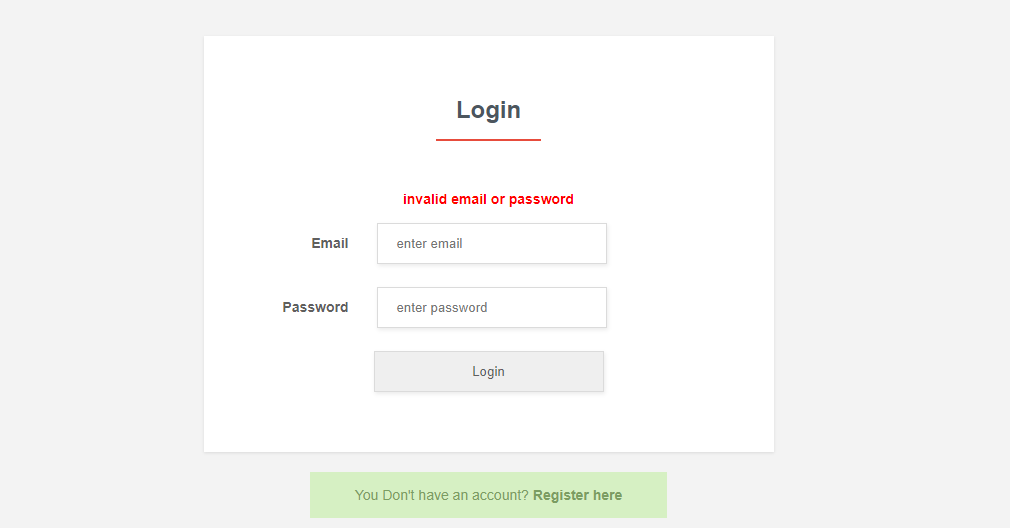
how the GUI flows.

## Jobseeker Module

A registered User logs in with correct login credentials and if he forgets his password a mechanism to retrieve the password is implemented.

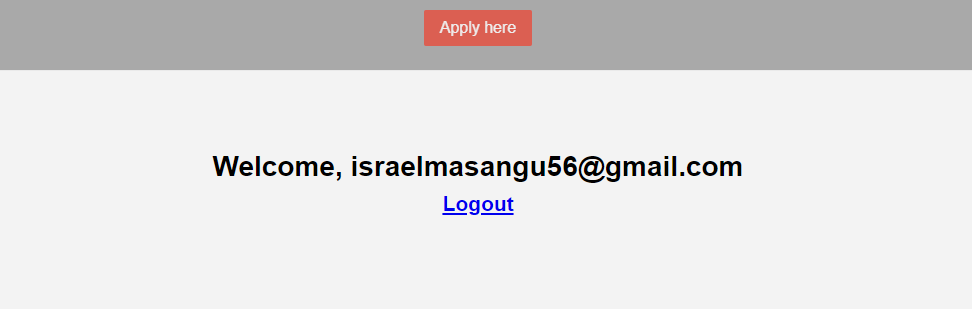


**Figure 8.1 Jobseeker Login Page**



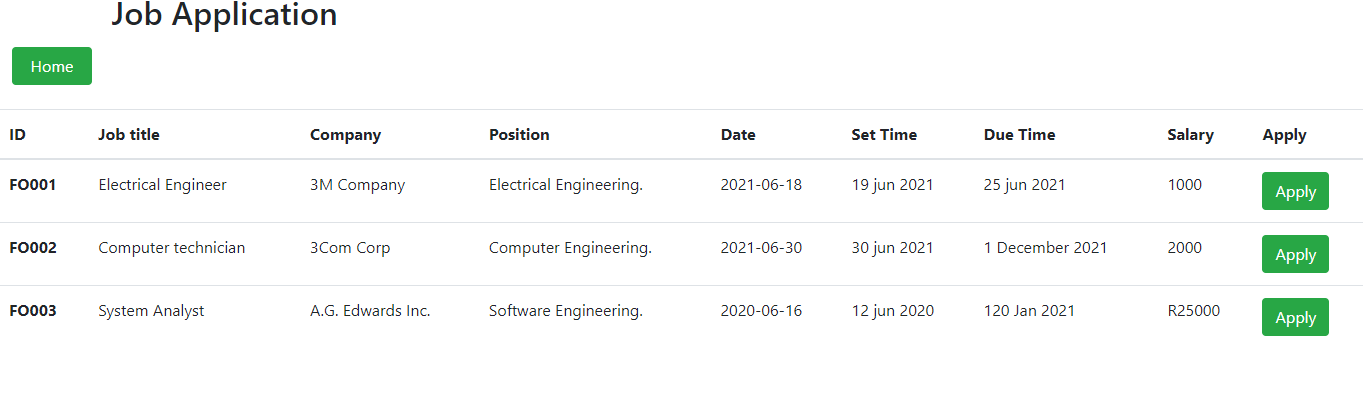
**Figure 8.2 Jobseeker Wrong Username/Password Error Message**

Figure 8.1 -8.3 shows the user login page and reflects the error shown when an incorrect password is given. It also shows the navigation screen for forgetting the password. Once the email id is given the password is forwarded in that mail id.

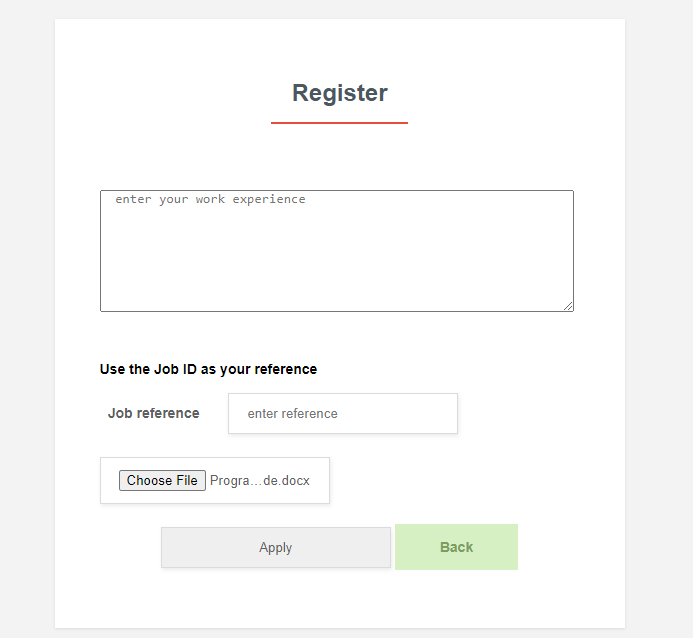


**Figure 8.4 Jobseeker Homepage**

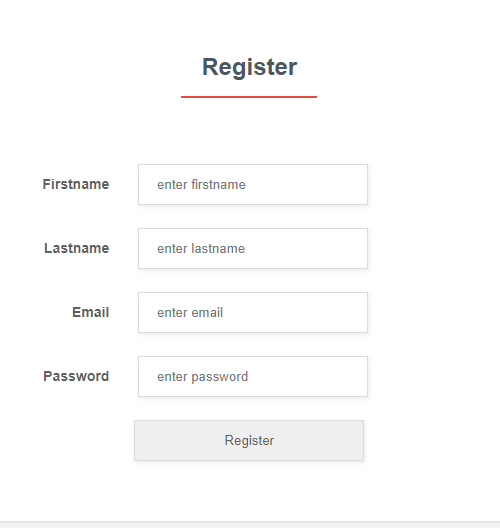
Once the user logs in to the account this is the user homepage which lists down all the jobs available and now the jobseeker can perform several activities like viewing all jobs, viewing applied jobs, upload their resumes, edit their profile information and logout finally.



**Figure 8.5 Jobseeker Applied jobs**

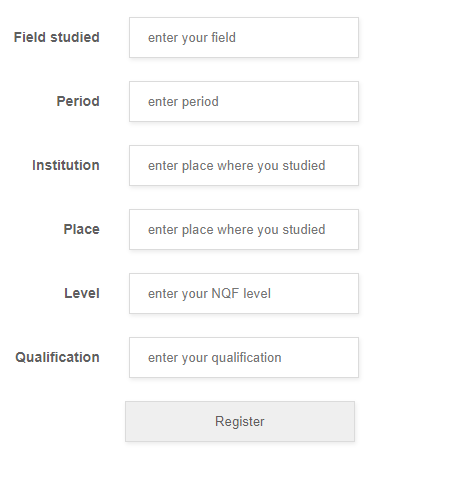
****

**Figure 8.6 Jobseeker Upload**



**Figure 8.8 Jobseeker Edit Profile Page**

Once this form appears a candidate can change any details like user name, email, mobile no, qualification or password. But if the user leaves any of the filed blank, an angular form error will be thrown to fill out the required field. It also handles the scenario that we can’t enter alphanumeric or alphabets for a mobile number. The password if changed is encrypted and saved in the database. Then this will be updated in the database with an update successful message.



**Figure 8.9 Jobseeker Profile Registration Page**

If the jobseeker is new to the portal, he must register himself to the portal to access the portal. The candidate must fill this form to be able to get registered.

# Chapter 9 - Conclusion

E-recruitment is one of the major advancements of the job industry today. Overcoming traditional methods of recruitment, e -recruitment has bought a revolutionary change in the world of interviews and recruitment. While this application aims in giving a user-friendly experience to the users with a simple but logical frontend it has achieved so at its completion. This application also achieves certain functional capabilities with the latest technology stack used in the industries today. The testing results shows that the application is scalable and can handle decent load. Also, this application does not have any geographical constraints as anyone from any part of the world can get registered to the application and search for jobs or post jobs.

Developing this project with a primary goal of learning new technologies, I have got immense exposure in understanding technologies like Java not only at the implementation level but also in understanding the background of such technologies. Similarly, I have also learnt AngularJS and have witnessed how powerful front-end tool can be to make your life a lot easier with front end developments. Some of the major challenges faced was in understanding the callback/promise concepts and implementing them in the application. To debug, test and run the application I have encountered many cutting-edge technologies and learnt about them. This invariably have enhanced my hands- on knowledge with a broad spectrum of technologies which would come handy once I start facing the industry after my graduation.

# Bibliography

1. Ganesh Kondal. (2014, June 13). *NodeJS - Server Side JS.* Retrieved on 10/10/2017 from LinkedIn. <https://www.slideshare.net/ganeshkondal/nodejs-server-side-js>
2. Features of Node.js Retrieved on 10/10/2017 from <https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm>
3. Rambabu Posa. (2015, May 30). *Node.js Components.* Retrieved on 10/10/2017 from JournalDev. [https://www.journaldev.com/7423/node-js-components-modules-npm-install-](https://www.journaldev.com/7423/node-js-components-modules-npm-install-update-uninstall-example) [update-uninstall-example](https://www.journaldev.com/7423/node-js-components-modules-npm-install-update-uninstall-example)
4. Eyal Vardi. (2013, May 12). *AngularJS Architecture.* Retrieved on 10/10/2017 from LinkedIn. <https://www.slideshare.net/EyalV/angularjs-architecture>
5. Harbinger Systems. (2015, May 12). *JavaScript MVC Frameworks: Backbone, Ember and Angular JS.* Retrieved on 10/10/2017 from LinkedIn. [https://www.slideshare.net/hsplmkting/java-script-mvc-frameworks-backbone-ember-and-](https://www.slideshare.net/hsplmkting/java-script-mvc-frameworks-backbone-ember-and-angular-js) [angular-js](https://www.slideshare.net/hsplmkting/java-script-mvc-frameworks-backbone-ember-and-angular-js)
6. Retrieved on 10/10/2017 from [https://programmaticponderings.files.wordpress.com/2015/01/mean-stack-architecture-general-](https://programmaticponderings.files.wordpress.com/2015/01/mean-stack-architecture-general-third-party-1.png) [third-party-1.png](https://programmaticponderings.files.wordpress.com/2015/01/mean-stack-architecture-general-third-party-1.png)