View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by K-State Research Exchange

Online Job Portal

by

Urmi Chakravarty

B. Tech., Netaji Subhash Engineering College, India, 2013

A REPORT

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computer Science
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2017

Approved by:

Major Professor
Daniel Andresen

Copyright

© Urmi Chakravarty 2017.

Abstract

“Dreams Job” is an online Job Search Portal, a web application through which job seekers can register and apply for jobs. Through this portal employers 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. Some of them are NodeJS, AngularJS, Sequelize ORM, etc.

Table of Contents

List of Figures vii
List of Tables ix
Acknowledgements x
Chapter 1 - Introduction 1
1.1 Project Description 1
1.2 Motivation 2
1.3 End users 3
Chapter 2 - Background and Technologies Used 5
2.1 Node.js 5
2.1.1. Node JS Architecture: 6
2.1.2: Advantage of Using NodeJS 7
2.1.3: Special features of NodeJS [2] 7
2.1.4: NodeJS Components: 7
2.2 Sequelize ORM 9
2.3 Angular JS 9
2.3.1: AngularJS Architecture 10
2.3.2: Special features of AngularJS 11
2.3.3 Angular JS Validation 12
2.3.4.MVC Architecture achieved through Angular 14
2.4 jQuery and HTML5 16
2.5 Node Server 16
2.6 MySQL 17
Chapter 3 - Related Work 18
3.1 Existing System 18
3.2 Proposed System 19
Chapter 4 - Requirement Analysis 20
4.1 Requirement Gathering 20
4.1.1 Functional Requirements 20
4.1.2 Non-Functional Requirements 21

4.2 Requirement Specifications 22
4.2.1 Software requirements: 22
4.2.2 Hardware Requirements 22
Chapter 5 - System Design 23
5.1 Use Case Diagram 23
5.1.1: Description of Use Case Diagram 24
5.2 Activity Diagram 25
5.2.1: Description of Activity Diagram 26
5.3 Sequence Diagram 27
5.3.1: Description of Sequence Diagram 28
Chapter 6 - Database Design 31
6.1 ER Diagram 34
Chapter 7 - Testing 36
7.1 Testing Levels 37
7.2 Tests performed on Job-Seeker Module 38
7.3 Tests Performed on Admin Module 40
7.4 Tests Performed on Company Module 41
7.5 Bugs Encountered 42
7.6 Load Testing 43
7.6.1 Hardware Configuration Used for Testing 44
7.6.2 Software Configuration Used 44
Chapter 8 - Implementation(GUI) 50
8.1 Jobseeker Module 50
8.2 Admin Module 57
8.3 Company Module 61
Chapter 9 - System Metrics 66
9.1 Coupling 66
9.2 LOC 66
Chapter 10 - Conclusion 68
10.1 UI Story 69
Chapter 11 - Future Work 70

Bibliography 71

List of Figures

Figure 2.1 Node JS Architecture [1] 6
Figure 2.2 NodeJS Components [3] 8
Figure 2.3 AngularJS Architecture [4] 10
Figure 2.4 Angular Built-in Form Validation 13
Figure 2.5 Angular Custom Validation 14
Figure 2.6 AngularJS MVC Architecture [5] 15
Figure 5.1 Use Case Diagram of Job Search Portal 24
Figure 5.2 Activity Diagram of Job Search Portal 26
Figure 5.3 Sequence Diagram of Job Search Portal [6] 27
Figure 6.1 Company Details Table 31
Figure 6.2 Job Details Table 32
Figure 6.3 User Details Table 33
Figure 6.4 User_Job Table 33
Figure 6.5 Entity Relationship Diagram 34
Figure 7.1 Load Testing for Job Search Portal on the same system 45
Figure 7.2 Load test Result for job Search portal with client and server on different system 46
Figure 7.3 Load test Result for Job Search Portal on the same system 46
Figure 7.4 Load test Result for job Search portal with client and server on different system 47
Figure 7.5 Load test Result for Job Search Portal on the same system 47
Figure 7.6 Load test Result for Job Search Portal on the same system 48
Figure 7.7 Load test Result for job Search portal with client and server on different system 48
Figure 7.8 Load test Result for job Search portal with client and server on different system 49
Figure 8.1 Jobseeker Login Page 50
Figure 8.2 Jobseeker Wrong Username/Password Error Message 51
Figure 8.3 Jobseeker Forgot Password Page 51
Figure 8.4 Jobseeker Homepage 52
Figure 8.5 Jobseeker Applied jobs 53
Figure 8.6 Jobseeker Upload CV 53
Figure 8.7 Jobseeker Upload CV Successful 54

Figure 8.8 Jobseeker Edit Profile Page 55
Figure 8.9 Jobseeker Profile Registration Page 56
Figure 8.10 Jobseeker Profile Registration Error 57
Figure 8.11 Admin Login Page 57
Figure 8.12 Admin Home Page 58
Figure 8.13 Admin View Companies Page 59
Figure 8.14 Admin Approve/Disapprove Companies Message 59
Figure 8.15 Admin Job Approvals Page 60
Figure 8.16 Company Login Page 61
Figure 8.17 Company Registration Page 61
Figure 8.18 Company View Jobs Page 62
Figure 8.19 Company View Applications Page 63
Figure 8.20 Company Post a Job Page 64
Figure 8.21 Company Edit Profile Page 65
Figure 9.1 Lines of Code on the server side 67
Figure 9.2 Lines of Code on the client side 67

List of Tables

Table 7.1 Tests performed on Job-Seeker Module 38
Table 7.2 Tests performed on Admin Module..... 41
Table 7.3 Tests performed on Company Module 42

Acknowledgements

I would like to thank my Major Professor, Dr. Daniel Andresen for believing in me, my capabilities and keeping his faith on me that I would be able to finish the project on time. His constant guidance, encouragement and valuable feedback led to the successful completion of this project.

I extend my heartfelt thanks to Dr. Torben Amtoft for serving on my committee and being my academic advisor, guiding me through all the major decisions and above all believing in my abilities.

I would also extend my heartfelt thanks to Dr. Mitchell Neilsen for taking time to serve on my committee and for being there forever for thousands of academic queries that you have helped me with and guided through some difficult decisions.

And I cannot finish without thanking my Mother 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

1.1 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. "Dreams Job" 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 search 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.

1.2 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 NodeJS, Sequelize ORM, AngularJS, Html, CSS, jQuery and MySQL as database and have gained enough experience and exposure working on them.

1.3 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 with NodeJS and its NPM package for the server side, Sequelize, a promise based object relational model, NodeJS web server and AngularJS as the front end with ES6 JavaScript, Html5, CSS3 and jQuery. All validations are done with the AngularJS. 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. JavaScript following both Object Oriented paradigm through prototypal inheritance and functional programming paradigm, working with JavaScript frameworks which includes NodeJS and Angular JS this application gave me immense exposure to these technologies.

2.1 Node.js

In traditional web applications, JavaScript is treated as a client-side coding language which enhances and extends the capabilities of client-side coding. However, NodeJS is a programming language that is run on the server side. It is JavaScript itself but enhanced with the capabilities to work as a server-side development language outside the browser. In this project, I have used NodeJS with its famous module Express to achieve server-side functionalities. NodeJS is built on googles V8 engine.

2.1.1. Node JS Architecture:

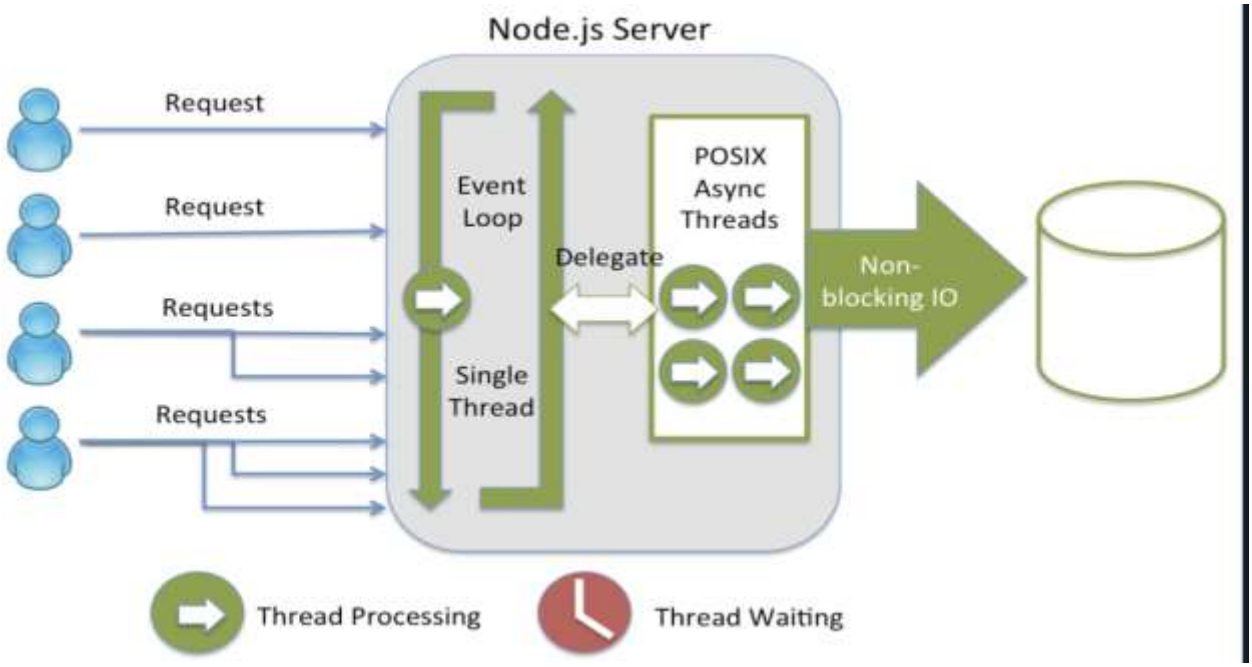


Figure 2.1 Node JS Architecture [1]

Architecture Description:

- User sends a request to the server.
- This request gets lined up in a queue of requests.
- A single thread, with an event loop mechanism, delegates each request to a thread in the thread pool.
- This delegation method depends on the type of operation that thread in the thread pool is handling like input/output operations.
- Once the thread in the thread pool is allocated with the request, it processes the request asynchronously i.e. it keeps on processing other requests too and does not

wait for this request to get fulfilled. When this request gets resolved it gives back a result/error. This mechanism of asynchronous processing is also called callback mechanism or promise.

2.1.2: Advantage of Using NodeJS

The major advantage in using Nodejs is that being single threaded in nature Nodejs eliminates the race /concurrency issues faced by multithreaded applications.

2.1.3: Special features of NodeJS [2]

1)Asynchronous and event driven - All the API of a node.js library is non-blocking in nature. After calling an API the server moves to the next API and does not wait for the response. Once the request is resolved a promise is returned to the server, which is a callback mechanism that lets the server know the result or the error. Hence NodeJS is fast.

2)Highly scalable - Nodejs follows a single threaded mechanism as we know with event looping. It is highly scalable because the callback/promise mechanisms help the server to respond in an asynchronous manner and makes it highly scalable in contrast to conventional client-server applications.

3)No Buffering - Nodejs cuts down the overall processing time by uploading a video or audio file. Node.js application never buffers any data it simply outputs data in chunks.

2.1.4: NodeJS Components:

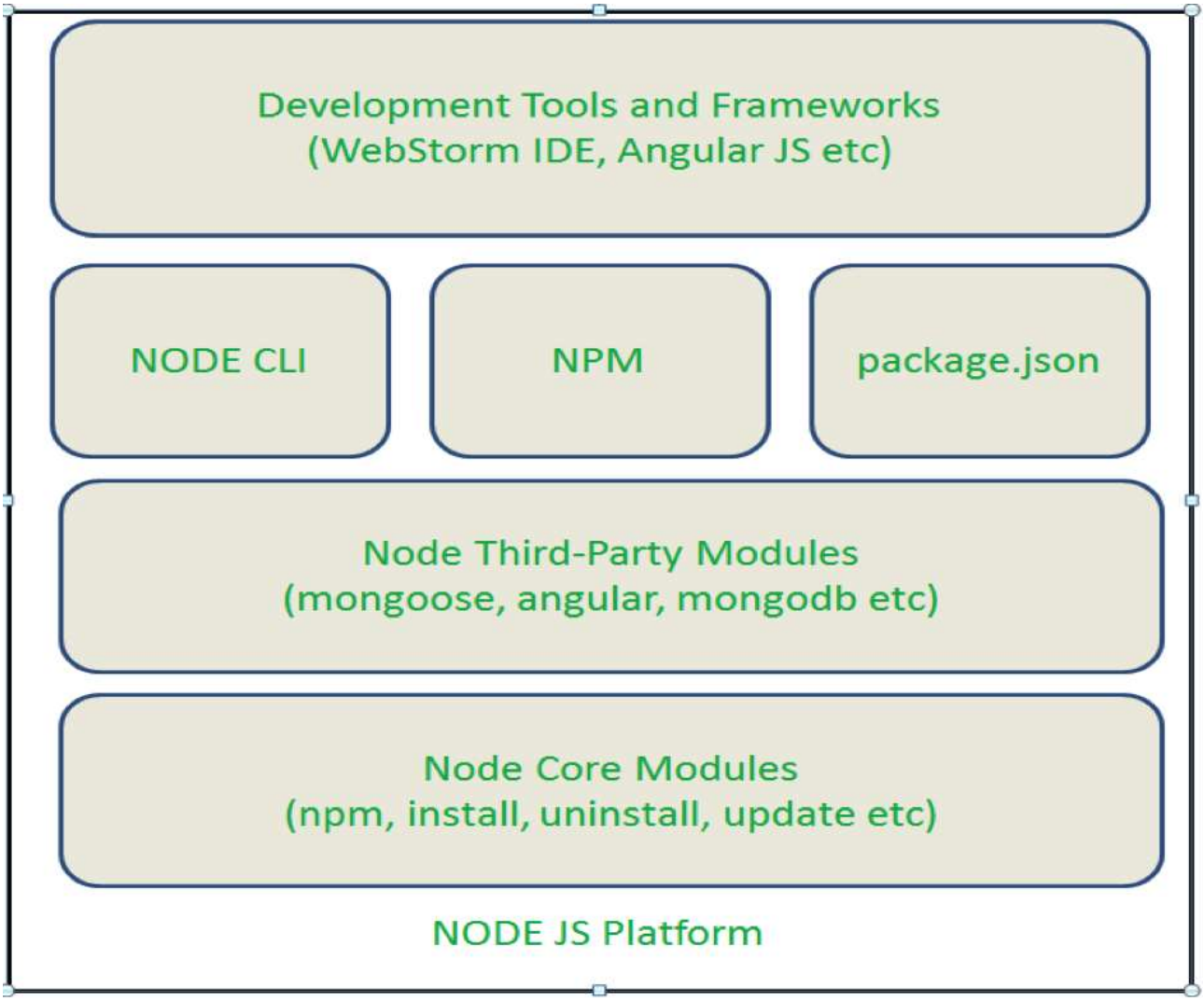


Figure 2.2 NodeJS Components [3]

The Node components include the Node Package Manager which contains all the node core modules as shown in the figure 2.2, it depends on third party modules as in my project Angular, it has a CLI which is installed by default, it has a Json formatted text file which contains all the important package details called package.json and finally it also depends on UI frameworks like in my case AngularJS.

2.2 Sequelize ORM

The process of mapping objects to RDMS i.e. Relational Database Management Systems is called Object Relational Mapping. Sequelize, a Node package, written in JavaScript, is a promise oriented ORM for Node.js that helps interaction with a variety of SQL databases including PostgreSQL, MySQL etc. For its promise features Sequelize is dependent on Bluebird Library which is a fully featured promise library. Features like synchronization, association, validation are all supported by Sequelize. Sequelize ORM acts as an interface between two different systems. Also, when we want to connect between different databases Sequelize provides great features to achieve so. Sequelize connects between this restful application and SQL database at the backend. We create an object that contains the schema definition and then expose this object by importing this object to apply specific operations on our required databases. To chain error and success callbacks, Sequelize uses promises and all these suits well with unit tests and this function chaining leads to increased readability and maintainability along the way.

2.3 Angular JS

Angular JS is an open source JavaScript based structural web framework that follows the MVC architecture. Using Angular we can load a HTML page and dynamically update the web page without constant reloading. Such web applications developed with the help of Angular are called Single Page Applications. SPA helps in eliminating the server-side load of rendering web pages as much of the work in this kind of application is handled by JavaScript in the client side. Angular helps in overcoming the pitfalls of Html that helps view web pages statically. Angular help in dynamic views of the webpages. We write

directives (we would discuss it shortly) to manipulate the Document Object Model in the controller. The beauty of Angular lies in the fact that the controller is unaware of the presence of directives and communications are attained through sharing the scope and $scope events.

2.3.1: AngularJS Architecture

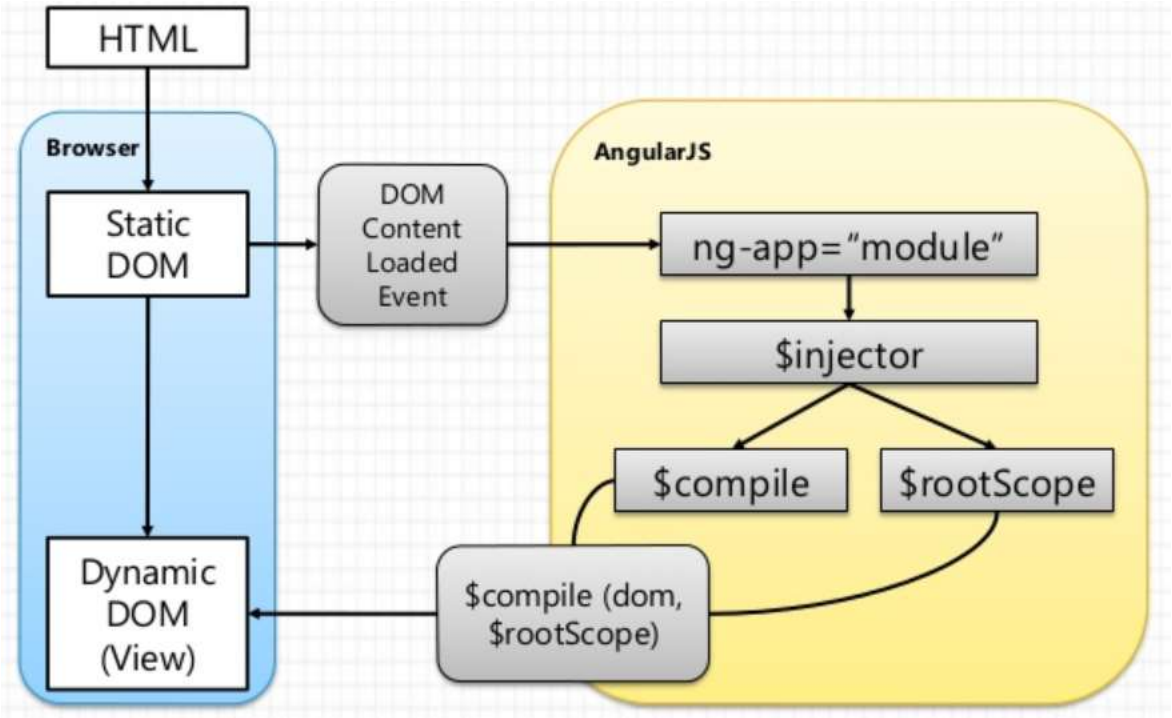


Figure 2.3 AngularJS Architecture [4]

Architecture Description:

At first, the web application is launched by bootstrapping a root module. Once an URL request is made by the web application, AngularJS will be downloaded, and the bootstrap callback gets registered. The callback gets executed once the HTML page gets totally downloaded. Angular then travels through the DOM tree structure to find the ng-app directive. Angular finds the element that contains the ng-app directive and declares it as

the root of the web applications DOM tree. An injector is created for Dependency Injection. For the context of the web application's model $rootscope object is created. Angular then walks down the DOM tree by compiling the DOM initiating at the ng-app root and next by processing directives and bindings.

2.3.2: Special features of AngularJS

1)**Two-way Data Binding** - In traditional data binding any impact on the model impacts the DOM and changes in the DOM makes it necessary to make changes in the model. And this reaches its complicacies with user interaction as the programmer then needs to detect the interaction and update the model and the view. But this is overcome by Angular's two-way data binding in which any change in the view gets reflected in the model and any change in the model gets reflected in the view. Both model and view are synchronized without the developer putting extra efforts to identify and modify the interactions. This is achieved by creating the ng-model directive that binds the model to the view.

2)**Directives** - Angular gives us the ability to build custom HTML tags called directives. Directives are a distinct feature that angular offers us. Directives mean decoration of HTML elements with new behaviors and manipulating attributes of HTML elements in an interesting manner. Directives works the same way as normal Html tags. Angular also has inbuilt directives like ng-app, ng-repeat, ng-controller, etc.

3)**Efficient templates** - The plain old HTML plus Angular's customized feature makes powerful templates for the web applications. In this Html elements are attached to angular

directives and expressions. These templates pretty much look like normal markup except the attributes. The view is made dynamic by adding a controller and a model at the view. The Directives, binding the view with model with curly braces and expressions, filters and form controls to validate inputs are what Angular templates are made up of.

4) **Dependency Injection -** If any Angular module or function is dependent on any inner Angular service, Angular provides a simple mechanism to get to use that service. All we need to do is to pass the service name as a parameter to the function that needs the service, i.e. we are injecting the dependency and angular provides an instance of the service. We do not need to write crucial codes/logic for this neither we need to search for these. This mechanism provided by Angular is called Dependency Injection.

2.3.3 Angular JS Validation

All input validations for this web application is handled by Angular i.e. the client side. Angular provides easy and simple solution to handle validations that makes the interface extremely user friendly as all the errors are immediately shown to user with instructions on how to handle them. When we write the Angular templates i.e. the view we use Angular form controls to validate user inputs. Angular uses directives to match the attributes in its templates with validator functions which are of two types a) Synchronous Validators b) Asynchronous Validators. The synchronous validator functions would take in a control instance and immediately return errors or null (if passed). The asynchronous validators on the other hand returns a promise. In the form control the synchronous validators are passed as the second arguments and the asynchronous as the third arguments.

Keeping performance in mind the asynchronous validators runs only if all the synchronous validator functions returns true i.e. passes.

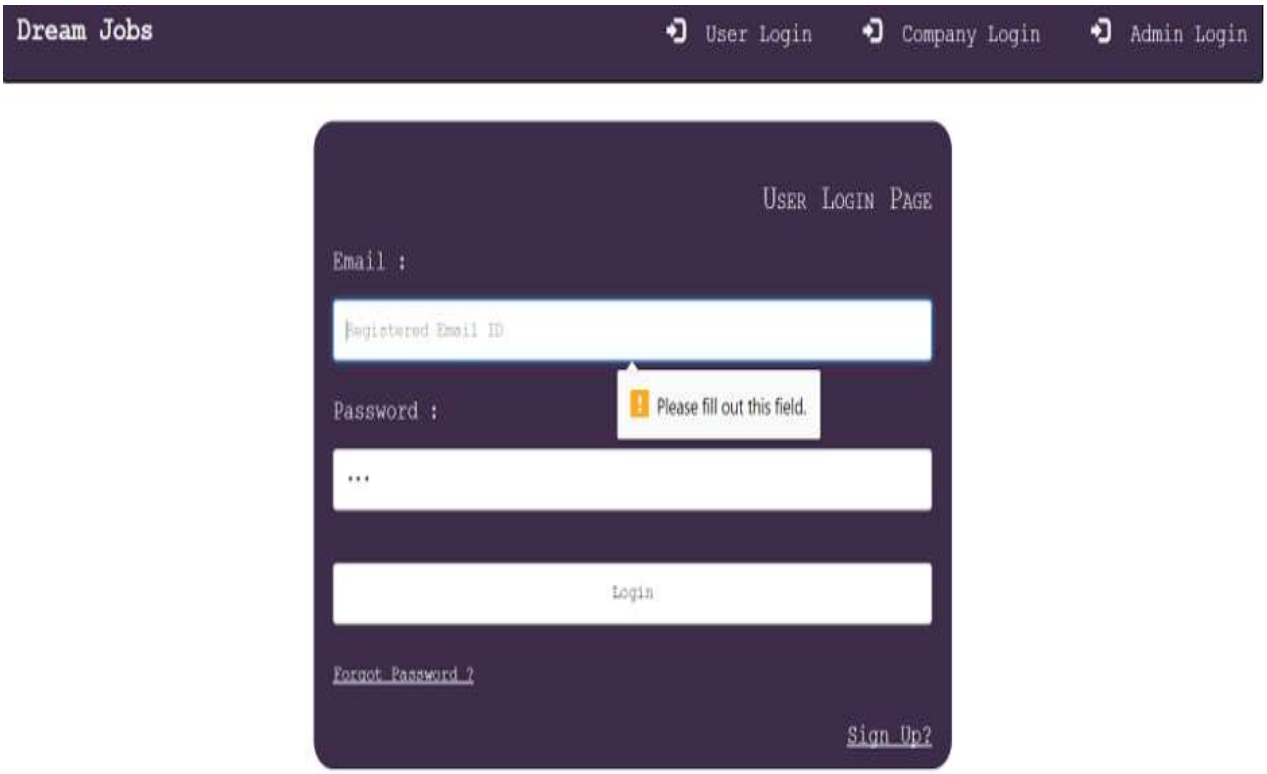


Figure 2.4 Angular Built-in Form Validation

Angular has built in validators as well as we can make customized validators. Whenever a user interacts with the interface the Angular form control highlights the field and displays a message if the input is not valid.

Figure 2.5 Angular Custom Validation

Angular provides a way to retrieve the current state of input fields as well as forms. Some of these states are- $untouched if the field in the form is not yet touched, $touched if the field has been touched, $pristine if the field is not yet modified, $dirty if the field is modified etc. All these states are returned in Boolean. In short Angular validations gives a rich user experience and makes it worth using Angular Framework.

2.3.4.MVC Architecture achieved through Angular

MVC is a software design pattern which is a 3 tier Architecture. MVC stands for Model View Controller.

Model - This handles all the data from the database.

View - This is the user interface or the front end.

Controller - This is where we have all the business logic i.e. something which binds the model and the view together.

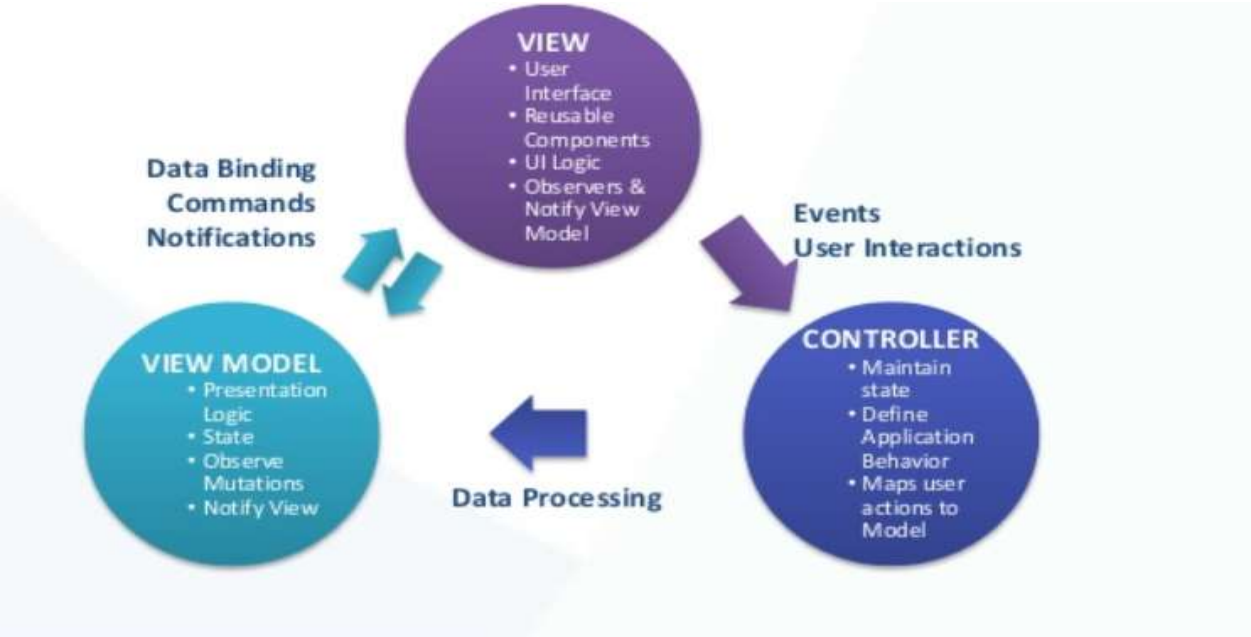


Figure 2.6 AngularJS MVC Architecture [5]

Model - The model is the client-side data. All data from the database comprises of the model. A model in Angular is simply a JavaScript object i.e. the $scope object. This model object is tied to a controller that receives data from the source and renders it to the view.

View - This is the Html templates along with Angular directives and Expressions. The view handles the layout and passes off any interaction by the user to the controller.

Controller - This is the business logic where angular performs two-way data binding. The controller defines the behavior and responds to user interactions accordingly. For this we use the inbuilt ng-controller directive.

Easy testing, flexibility, customizability, readability and open source community makes Angular JS worth a technology to use and have experience of.

2.4 jQuery and HTML5

jQuery is a JavaScript library that enhances and makes client-side coding more powerful along with Html. Handling events, Asynchronous JavaScript actions, traversing the Html and manipulating the DOM is a cake walk courtesy jQuery. jQuery enables one to handle critical UI functionality using very less code which is simple and cleaner. jQuery is extensible as it can be extended to include customized behavior. It is fast and improves the performance of the application.

2.5 Node Server

The most commonly used web server on Node is Express. Node is the runtime environment for web applications in JavaScript. NodeJS creates its own server. ‘http.createServer(app).listen(9000)’ this template makes it quite clear that how Node first creates its own server and listens to it to a specified port. when Node gets a CPU intensive request calls, each other request gets blocked till this CPU intensive request stops for an I/O. It is a great functionality of Node to delegate CPU intensive requests to some other process. Nodejs handles all HTTP requests with its core module http. Node is very faster for performing input and output operations and it can handle a huge number of concurrent connections with high throughput, which equates to very high scalability.

2.6 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

3.1 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.

3.2 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

4.1 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.

4.1.1 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 search based on advanced search
- Advanced search can be done on role, salary, technology, etc.
- 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.
- Job seeker can retrieve their password if forgot.
- Companies can Register to the portal.
- 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.

4.1.2 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.

4.2 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:

4.2.1 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, AngularJS, JavaScript, jQuery
- Web Service: RESTful web services
- Browser: Preferable Google Chrome or Mozilla Firefox

4.2.2 Hardware Requirements

- Processor: Intel core i5
- Processor Speed: 2.50 GHz
- RAM: 8 GB

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.

5.1 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.

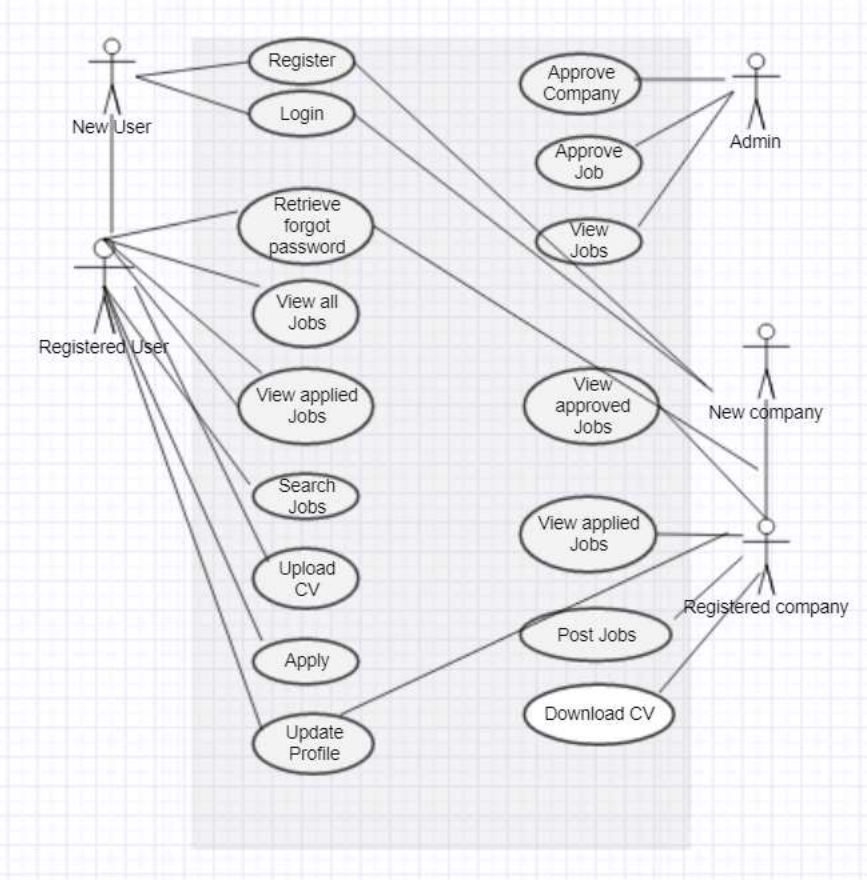


Figure 5.1 Use Case Diagram of Job Search Portal

5.1.1: Description of Use Case Diagram

In this system there are 5 actors namely admin, new user, registered user, new employer, registered employer. The different use cases are:

Approve Company – Once a company registers itself to the job portal, it must wait for the admin to approve or in other words give it permission to be a part of the portal.

Approve Job – Once the company posts a job, it should wait for the admin to approve the job before it is reflected on the portal.

View Jobs-The admin can view all the jobs on the portal.

Register – If there is a new User or a Company they first must sign up to the job portal.

Login- Once the new company or the new user signs up they can sign in to the portal.

Retrieve Forgot Password – If the registered user or the registered company forgets their password, they can retrieve their passwords through e- mails.

View all jobs -Once the user logs in to his/her account, they can view all the available jobs.

View applied jobs -Once the user logs in, he can view all his applications.

Search Jobs - User can do advanced search to search for specific jobs with his specific requirements.

Upload CV- User can upload their resume in specified file format to this job portal.

Apply -Users can apply to desired jobs.

Update profile - Users or Companies can edit their profile information.

View approved jobs – Company can view the jobs approved by the admin.

View applied jobs -Company can view the applications.

Download CV- Company can also download resumes and check applications.

5.2 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.

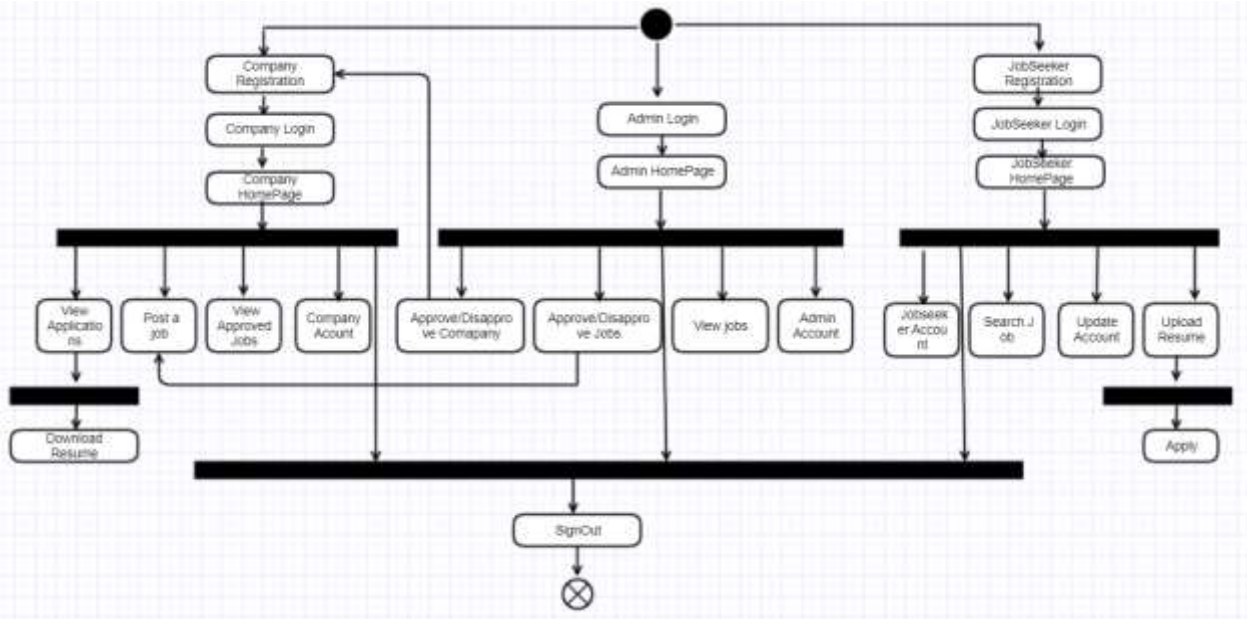


Figure 5.2 Activity Diagram of Job Search Portal

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.

5.3 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 andfunctions.

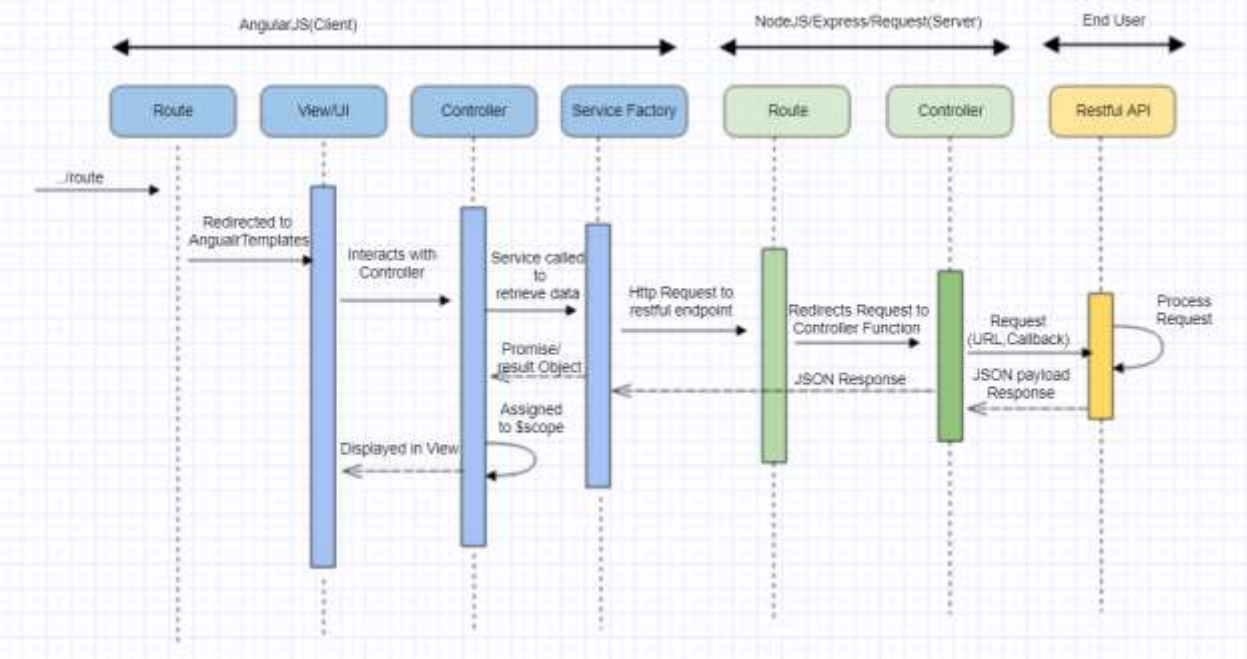


Figure 5.3 Sequence Diagram of Job Search Portal[6]

5.3.1: Description of Sequence Diagram

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

Client Side (Angular JS)

There are altogether 14 view templates used for this system. The view templates are:

Admin_login.html -interacts with the **Admin Controller** named as **admin_app**

Company_login.html-interacts with the **Company Controller** named as **company_app**

CompanyEditProfile.html-interacts with the Company Controller named as company_app

Job approval.html- interacts with the Admin Controller named as admin_app

PendingCompanies.html- interacts with the Admin Controller named as admin_app

PostJob.html- interacts with the Company Controller named as company_app

UploadCV.html- interacts with the **User Controller** named as **user_app**

User_login.html - interacts with the User Controller named as user_app

User_profile.html- interacts with the User Controller named as user_app

View_applications.html-interacts with the Company Controller named as company_app

View_companies.html-interacts with the User Controller named as user_app

Viewappliedjobs.html- interacts with the User Controller named as user_app

Viewapprovedjobs.html- interacts with the Company Controller named as company_app

Viewjobs.html– interacts with admin Controller, user controller and company controller.

The **app.js** is the service factory which returns a promise or the resulting object to the view. So basically, as the user requests a page or manipulates the page the AngularJS interacts

with the controller logic and the service factory logic through several functions and methods and returns the result to the view and then to the user. The view interacts with all the controllers and the service factories mentioned above.

Server Side(NodeJS)

While the user interacts with the frontend to request data, the frontend interacts with the backend to fetch the data and deliver it to the end user. On the server side we have the model, which is basically the ORM that fetches data from the database and delivers it to the server-side controller. There are three server-side controllers namely company.js, user.js and jobs.js. Each controller has specific routes with logic mentioned in them which are basically http routes. When a user requests a route, the controller in the frontend invokes the controller in the backend which in turn return the specific routes to the view/template of the front end which the user sees.

Company Controller Routes:

- *FetchCompaniesByStatus*
- *saveCompany*
- *authenticateCompany*
- *updateCompany*
- *fetchCompanyById*
- *fetchJobSeekersbyCompanyId*
- *fetchCompanyByUserId*
- *VerifyUserorComapny*
- *sendEmail*

Job Controller Routes:

- *SaveJob*

- *fetchJobsByStatusAndCompany*
- *dropJob*
- *fetchJob*
- *updateJob*
- *fetchJobforSeeker*
- *fetchAppliedjobsforSeeker*
- *sendEmail*

User Controller Routes:

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

For all the SendEmail functionality a widely known Node module called the NodeMailer module has been used.

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:

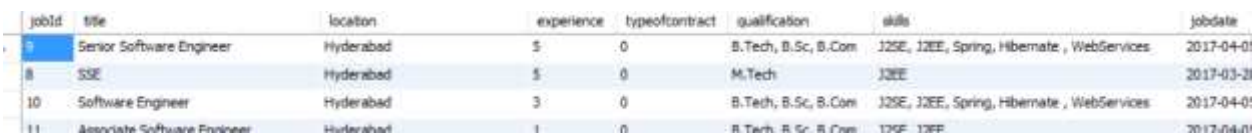
Company: It stores details about the company that registers itself to the portal and is approved by the admin/disapproved by the admin by a special flag column called status.

companyId	companyName	email	password	website	mobileno	status
1	Constient Global Solutions	hr@cgs.com	D#¥W□ÏÝn§□p•Ž[gm	http://cgs.com/	9848012345	1
2	Infosys	hr@infosys.com	»Ûakëß□ËhÇ□,ÚÇ	http://infosys.com/	9848012345	1
4	ABC IT Solutions	hr@abc.com	D#¥W□ÏÝn§□p•Ž[gm	http://www.abcit.com/	9848012345	1

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.



jobId	title	location	experience	typeoftcontract	qualification	skills	jobdate
9	Senior Software Engineer	Hyderabad	5	0	B.Tech, B.Sc, B.Com	J2SE, J2EE, Spring, Hibernate , WebServices	2017-04-0
8	SSE	Hyderabad	5	0	M.Tech	J2EE	2017-03-2
10	Software Engineer	Hyderabad	3	0	B.Tech, B.Sc, B.Com	J2SE, J2EE, Spring, Hibernate , WebServices	2017-04-0
11	Associate Software Engineer	Hyderabad	1	0	B.Tech, B.Sc, B.Com	J2SE, J2EE	2017-04-0

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, companyId 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.

userId	username	email	mobileNo	qualification	password	isCVUploaded
1	sd	dasdq@ed.com	0	cewf	}ÔQöùÂœ¤ªÁ'æ□□‹N	0
2	urmi	urmi@reddit.com	0	Ms	‹G...¥Šžñ#÷k§¶¿,□¬	0
3	swapnil	swapnil@quora.com	0	PhD	ß"GL@1YÊ‡ŽFË_¯¬	0
4	radhika	radhika21@gmail.com	0	Bcom	)NIÇ÷;^˜¢ESAÓã	0

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.

userId	jobId
1	11
8	8
11	11

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.

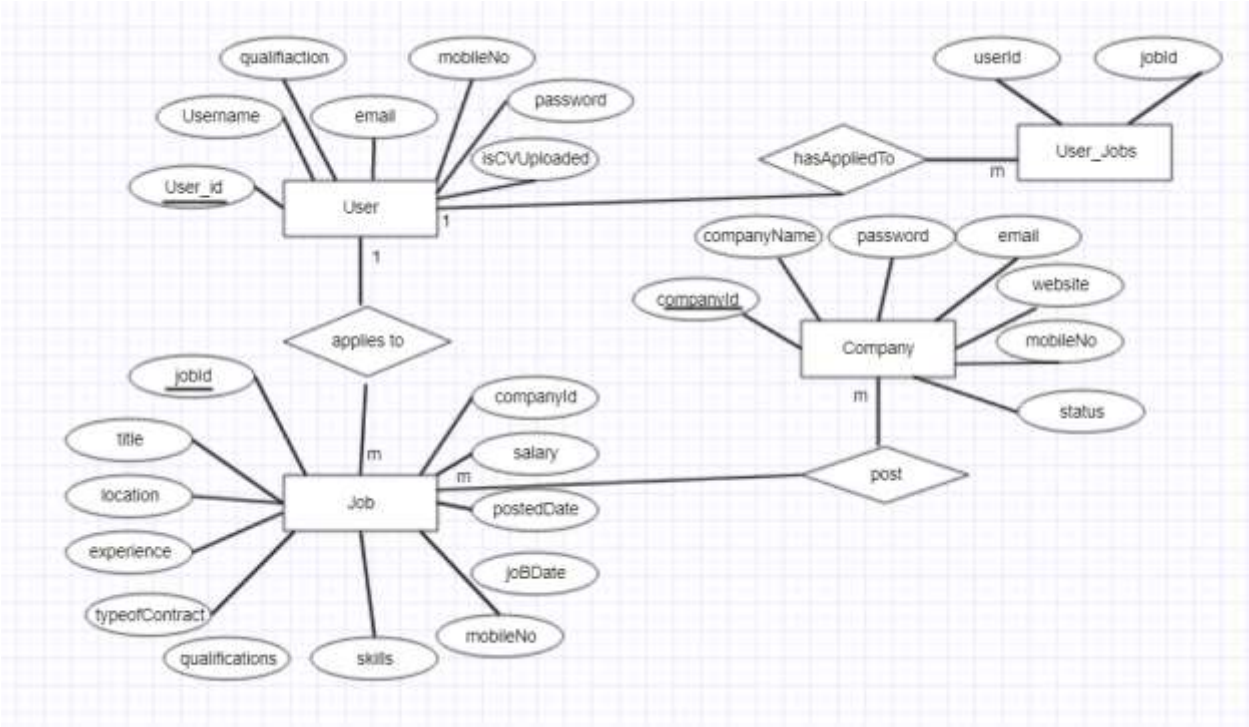


Figure 6.5 Entity Relationship Diagram

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 NodeJS 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.

7.1 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

7.2 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	Registers with existing credential	User /email already exists	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 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 advanced search button	Results in different searches selected	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

7.3 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 pending approvals button	Results in showing all the companies waiting to appear on the portal	PASS
Admin	Approve companies	Results in approval mail to companies	PASS
Admin	Disapprove companies	Results in disapproval mails to companies	PASS
Admin	Click the job approvals button	Results in showing all the jobs waiting to appear on the portal	PASS
Admin	Approve jobs	Results in approval mail to companies	PASS
Admin	Disapprove jobs	Results in disapproval mails to companies	PASS

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

Table 7.2 Tests performed on Admin Module

7.4 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	Clicks login button to see user homepage	User Homepage with all jobs listed is available	PASS
Company	Forgets password action	Receive email with password	PASS
Company	Misses filling a field in registration form	Receive error message to fill the blank	PASS
Company	Enter 10-digit mobile number for registration	Receive error message if not entered.	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	Views application	Can download CV	PASS
Company	Views jobs posted	Can delete jobs	PASS
Company	Clicks sign out	Logged out of the system	PASS

Table 7.3 Tests performed on Company Module

7.5 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.

7.6 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 loadtest. We can perform the loadtest 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. Loadtest also allows us to set requests per second.

7.6.1 Hardware Configuration Used for Testing

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

7.6.2 Software Configuration Used

- NodeJS version 3.10.10
- AngularJS version 4.0.0
- MySQL version 5.7

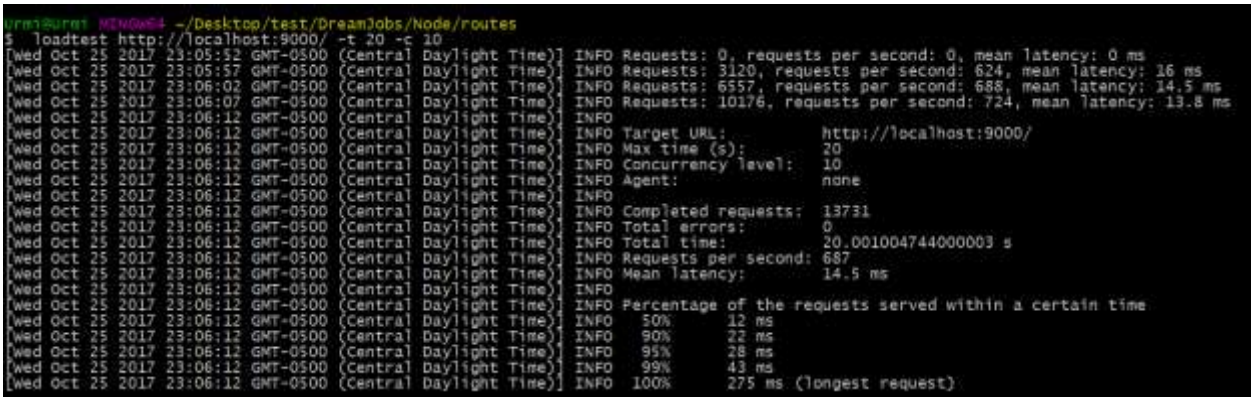


Figure 7.1 Load Testing for Job Search Portal on the same system

Here we have set the time limit t to 20 which represents the maximum number of seconds the server would wait until the requests no longer goes out and concurrency 10 which represents how many requests concurrently arrive at the server. Under such conditions the results represent that our server can handle 687 requests per second with a mean latency of 14.5ms. A percentage representation speaks that the server can process 50% of all the request within 12 milliseconds which is quite fast and effective. While the server can handle 90% request within 22 milliseconds to handle the last 5-10 % requests it requires about 28-43ms. The worst time required is 275ms.

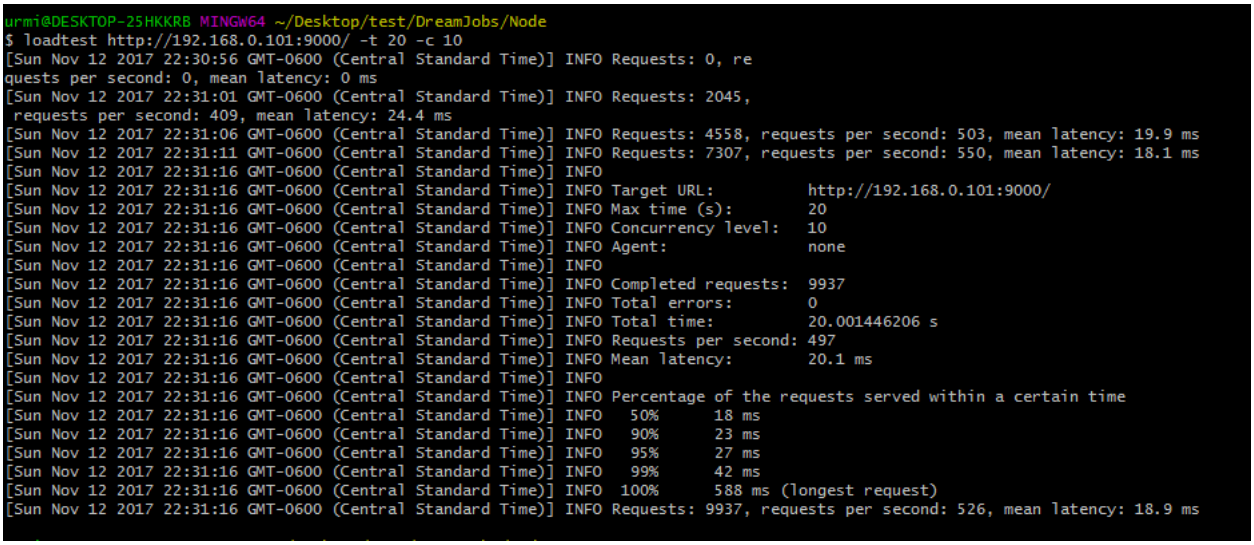


Figure 7.2 Load test Result for job Search portal with client and server on different system

Figure 7.2 shows that while I have performed load testing across different systems, the number of requests processed per second were 497with a mean latency of 20.1ms and the total completed requests were 9937 without any error.

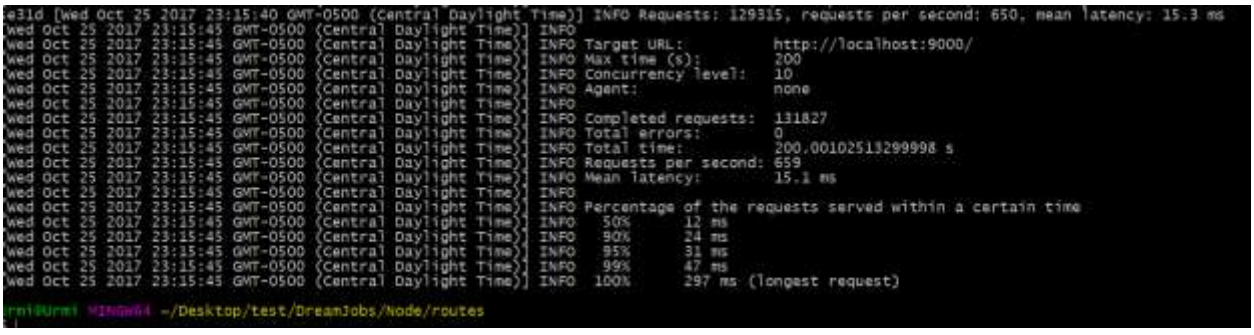


Figure 7.3 Load test Result for Job Search Portal on the same system

I have then increased the time limit to 200 and observed the results. It showed now the server can handle 659 requests per second with a mean latency of 15.1ms.

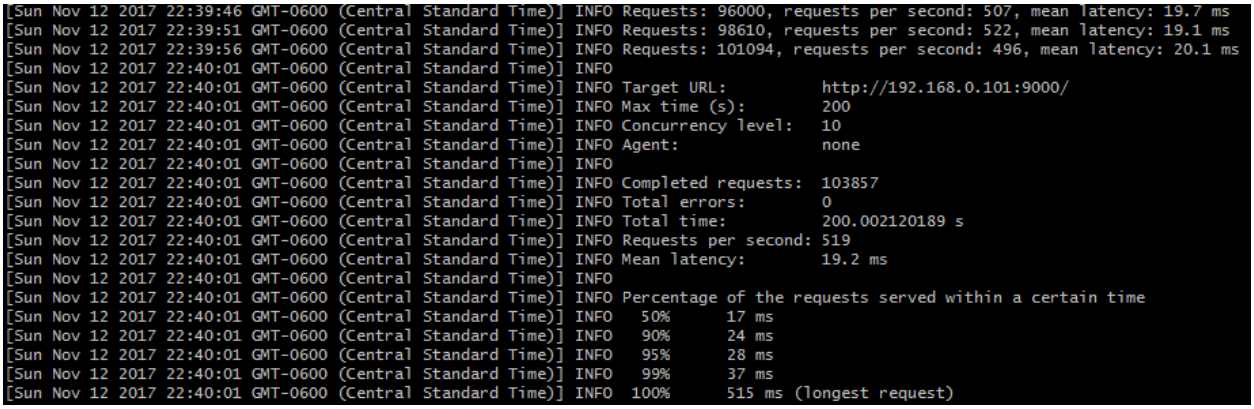


Figure 7.4 Load test Result for job Search portal with client and server on different system

Observing the result across different system showed that the server at a time of 200sceonds with 10 concurrent users can handle 519 requests per second with a mean latency of 19.2ms. The longest or worst time required to process the request was 515ms.

I wanted to see the consistency in the number of requests the server can handle, which seems to be nearly the same when we test this application with constant concurrent users and increasing the time limit until the requests no longer goes out.

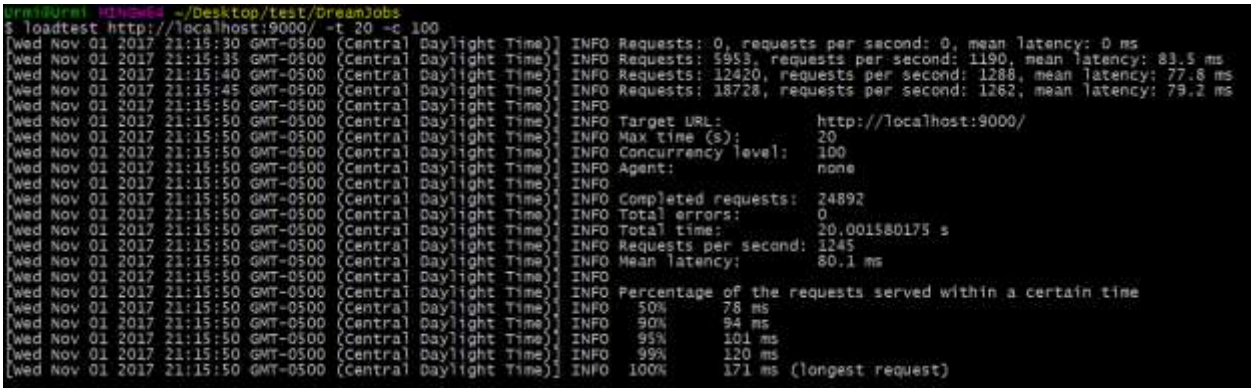


Figure 7.5 Load test Result for Job Search Portal on the same system

Figure 7.6 Load test Result for Job Search Portal on the same system

Keeping the time limit constant and increasing the concurrency we can see that the requests per second remains almost constant. The limiting factor was a concurrency of 10,000 where only 77890 requests were processed and the remaining failed.

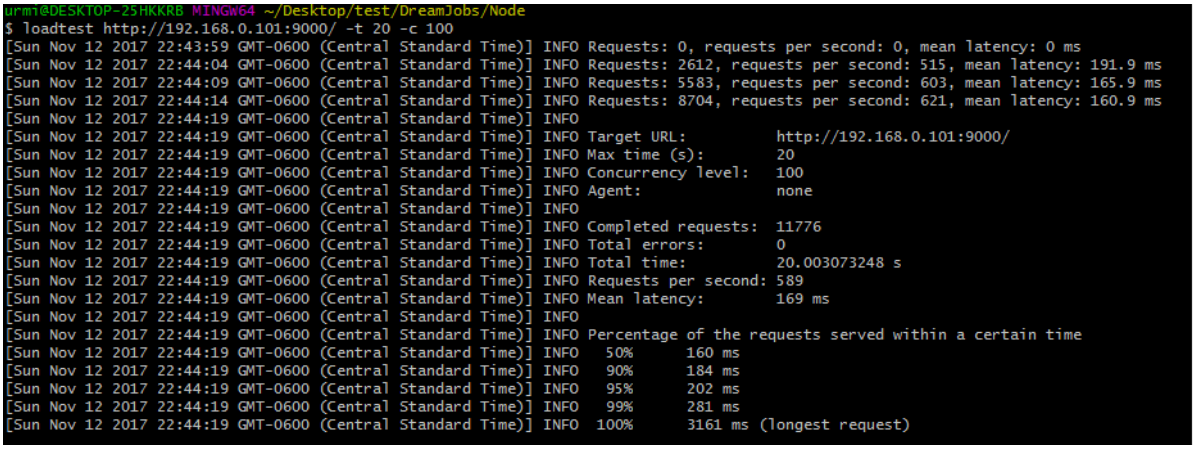


Figure 7.7 Load test Result for job Search portal with client and server on different system

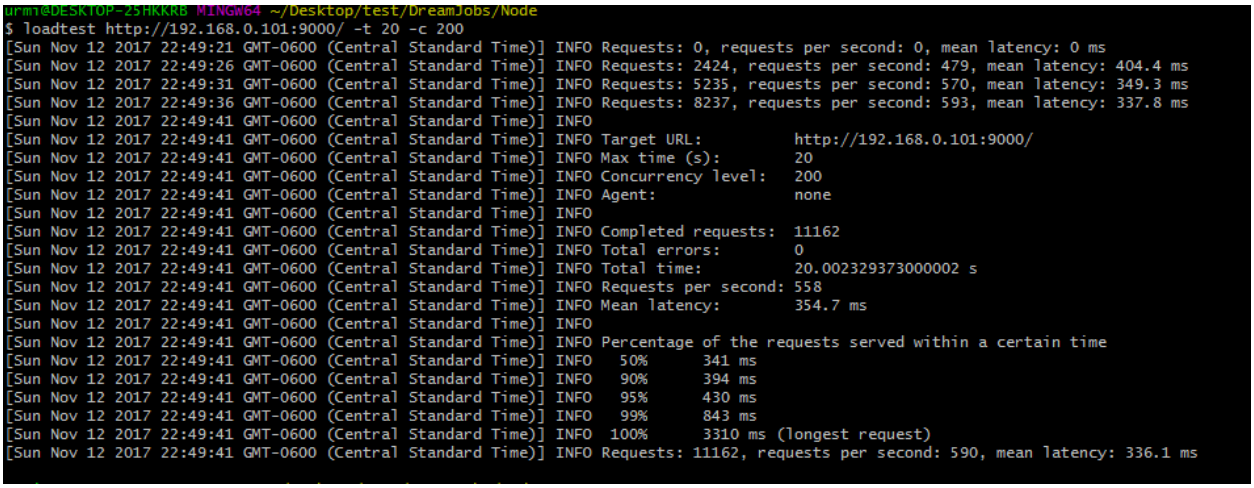


Figure 7.8 Load test Result for job Search portal with client and server on different system

Performing the load testing across different systems I observed that the requests processed per second remains almost same whereas the longest processing times are around 3161ms -3310ms.The limiting factor was a concurrency of 6000 this time where only 53664 requests were processed and the remaining failed.

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.

8.1 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.

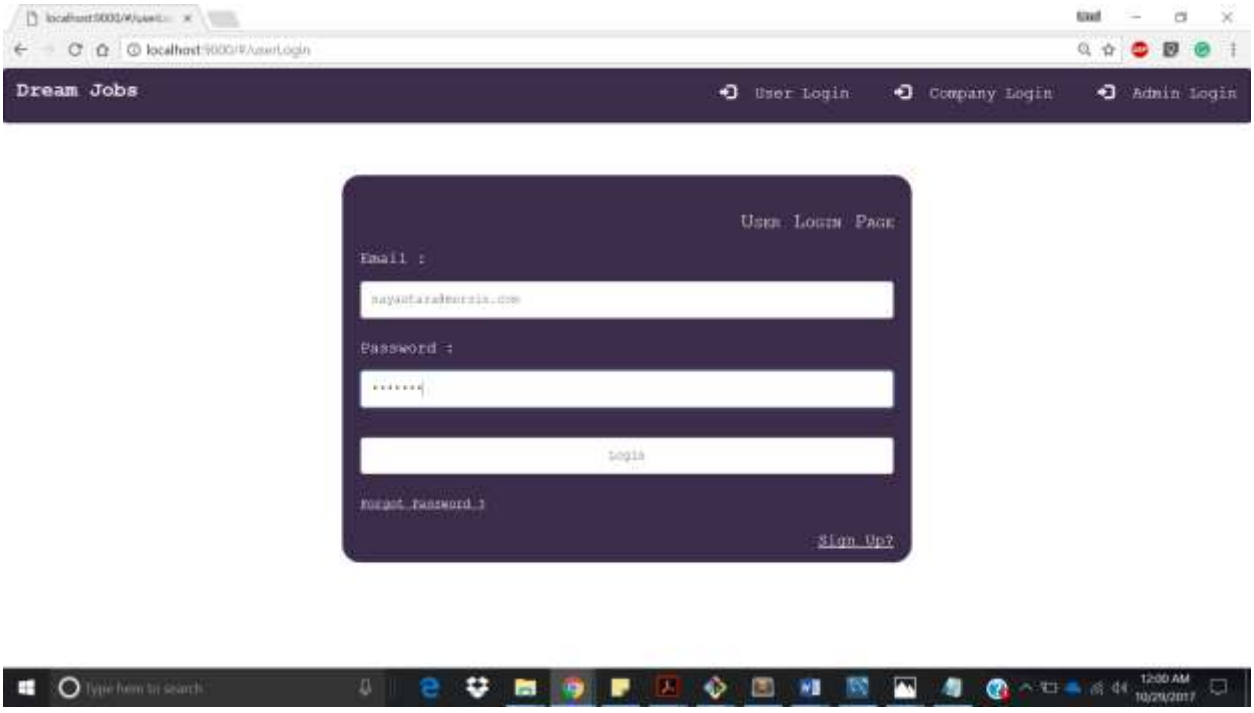


Figure 8.1 Jobseeker Login Page

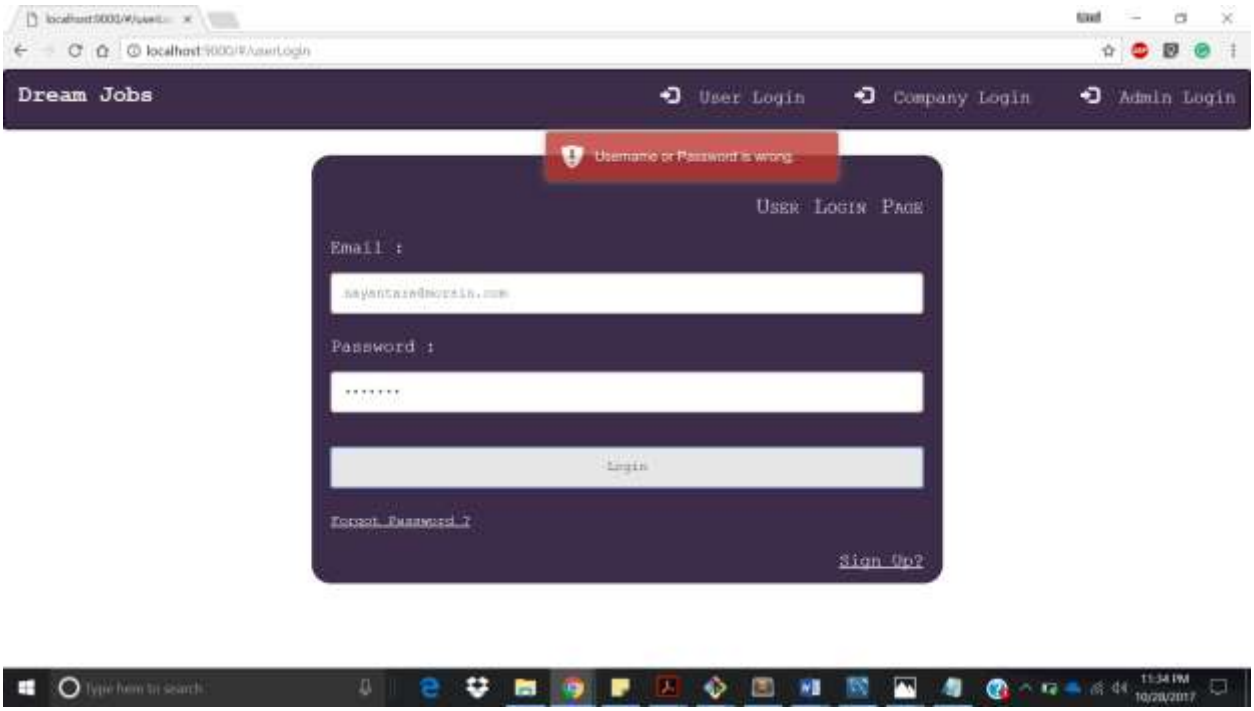


Figure 8.2 Jobseeker Wrong Username/Password Error Message

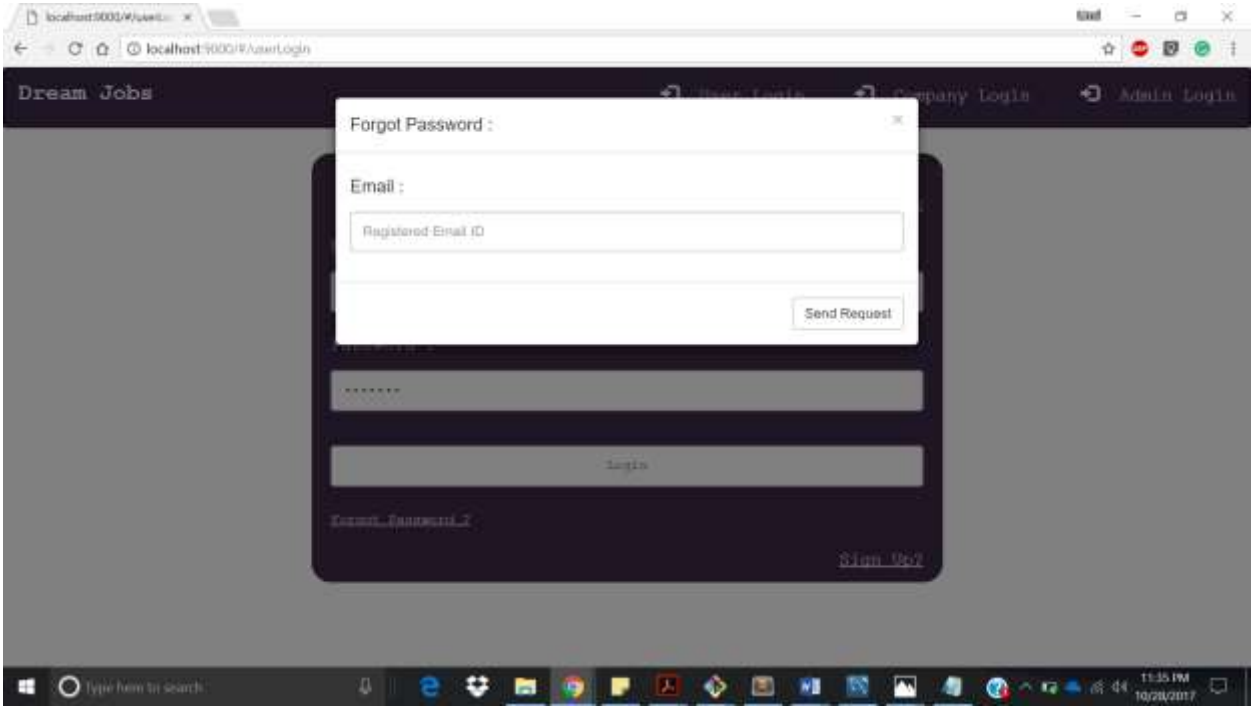
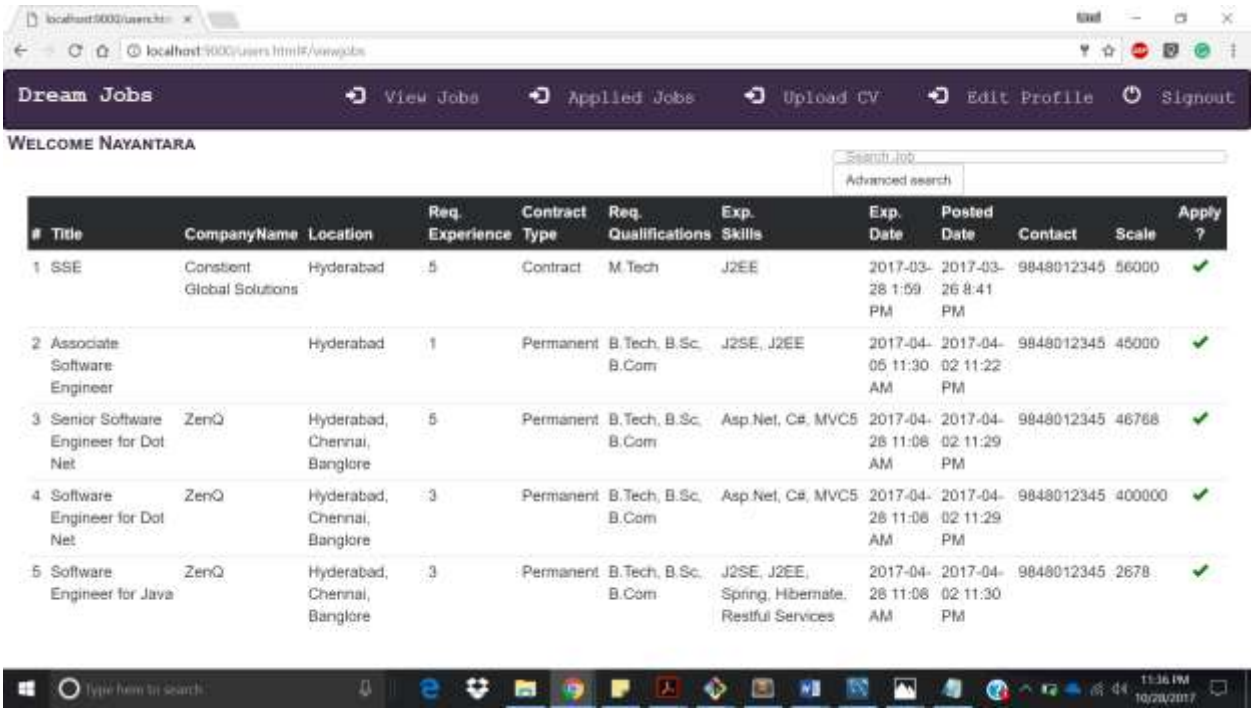


Figure 8.3 Jobseeker Forgot Password Page

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.



#	Title	CompanyName	Location	Req. Experience	Contract Type	Req. Qualifications	Exp. Skills	Exp. Date	Posted Date	Contact	Scale	Apply ?
1	SSE	Constient Global Solutions	Hyderabad	5	Contract	M.Tech	J2EE	2017-03-28 1:59 PM	2017-03-26 8:41 PM	9848012345	56000	✔
2	Associate Software Engineer		Hyderabad	1	Permanent	B.Tech, B.Sc, B.Com	J2SE, J2EE	2017-04-05 11:30 AM	2017-04-02 11:22 PM	9848012345	45000	✔
3	Senior Software Engineer for Dot Net	ZenQ	Hyderabad, Chennai, Banglore	5	Permanent	B.Tech, B.Sc, B.Com	Asp.Net, C#, MVC5	2017-04-28 11:08 AM	2017-04-02 11:29 PM	9848012345	46768	✔
4	Software Engineer for Dot Net	ZenQ	Hyderabad, Chennai, Banglore	3	Permanent	B.Tech, B.Sc, B.Com	Asp.Net, C#, MVC5	2017-04-28 11:08 AM	2017-04-02 11:29 PM	9848012345	400000	✔
5	Software Engineer for Java	ZenQ	Hyderabad, Chennai, Banglore	3	Permanent	B.Tech, B.Sc, B.Com	J2SE, J2EE, Spring, Hibernate, Restful Services	2017-04-28 11:08 AM	2017-04-02 11:30 PM	9848012345	2678	✔

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.

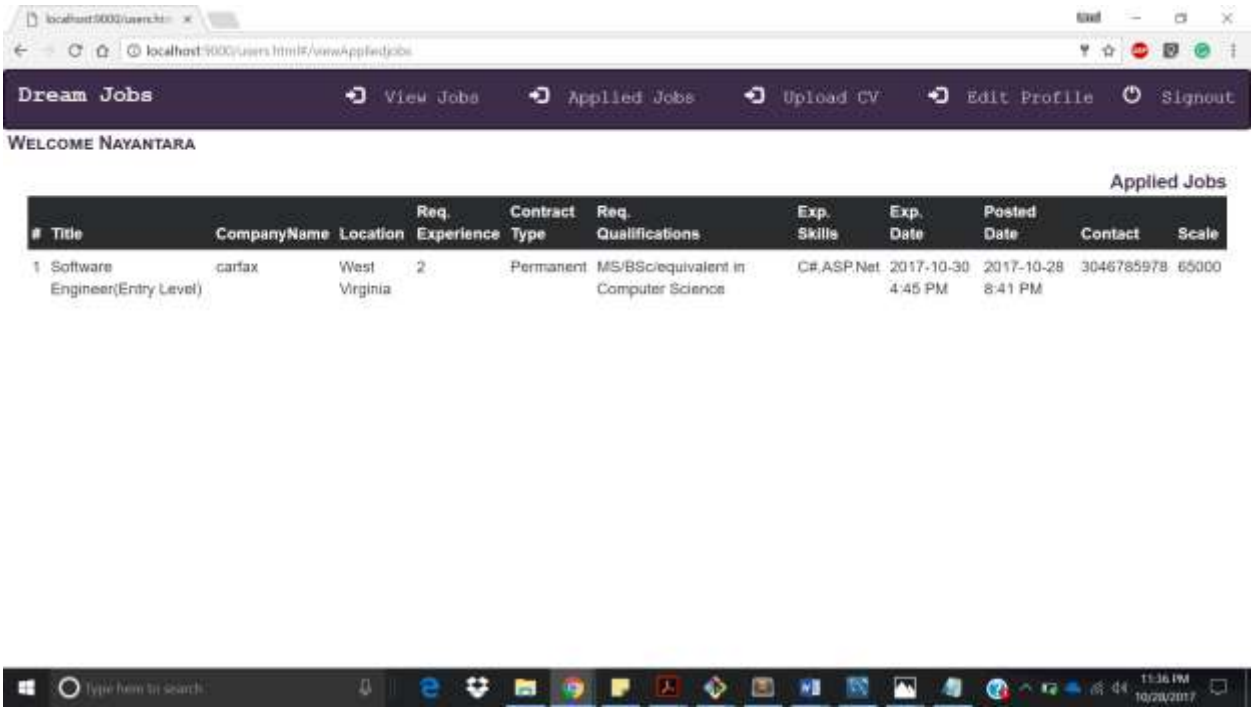


Figure 8.5 Jobseeker Applied jobs

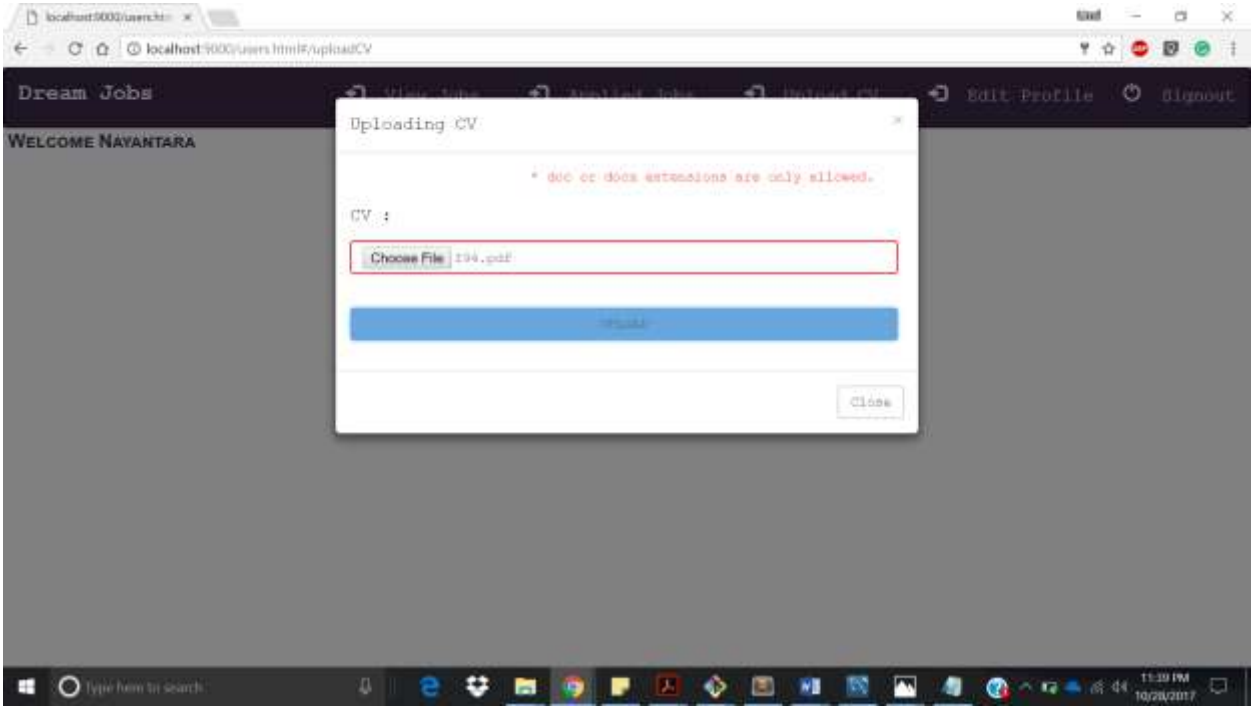


Figure 8.6 Jobseeker Upload CV

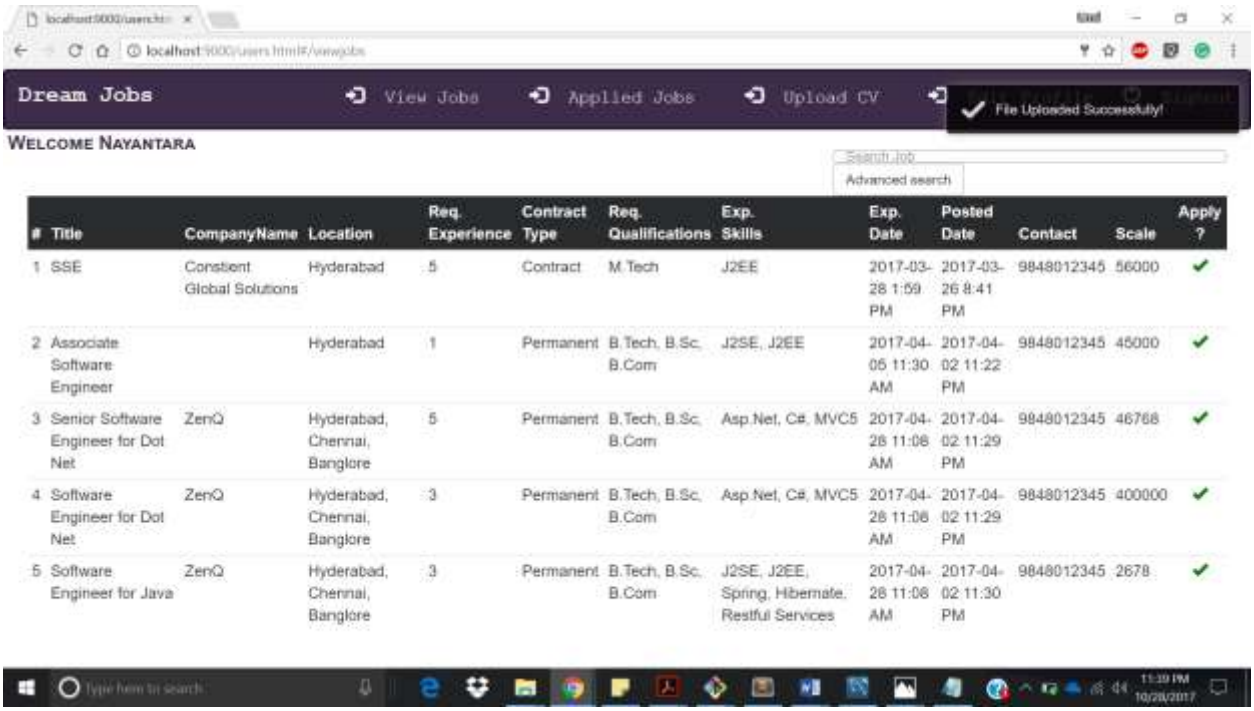


Figure 8.7 Jobseeker Upload CV Successful

Figure 8.6 and figure 8.7 firstly shows the modal box to upload CV with required file format. In figure 8.6 we can see that we tried to upload a file with .pdf extension and then we receive error. The upload box is highlighted in red. Once we give the proper file extension and click the upload button, the upload event takes place and a file upload success message is shown on the right hand top corner as we can see in figure 8.7.

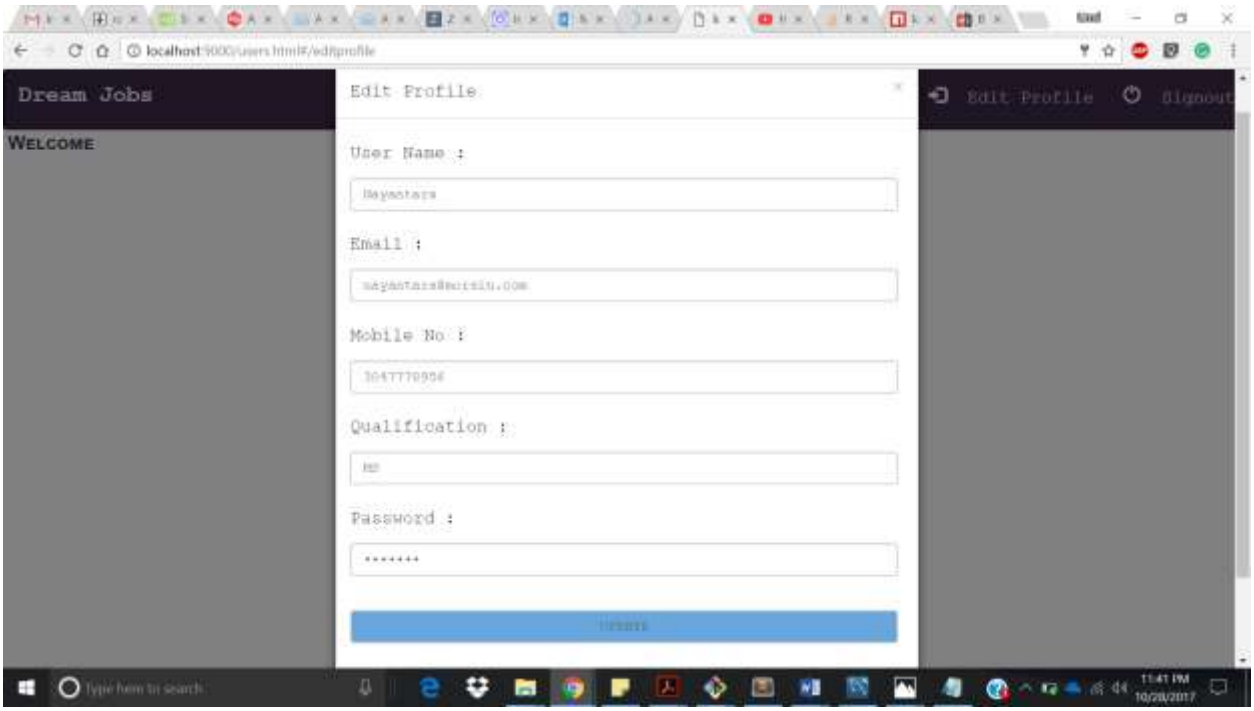


Figure 8.8 Jobseeker Edit Profile Page

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.

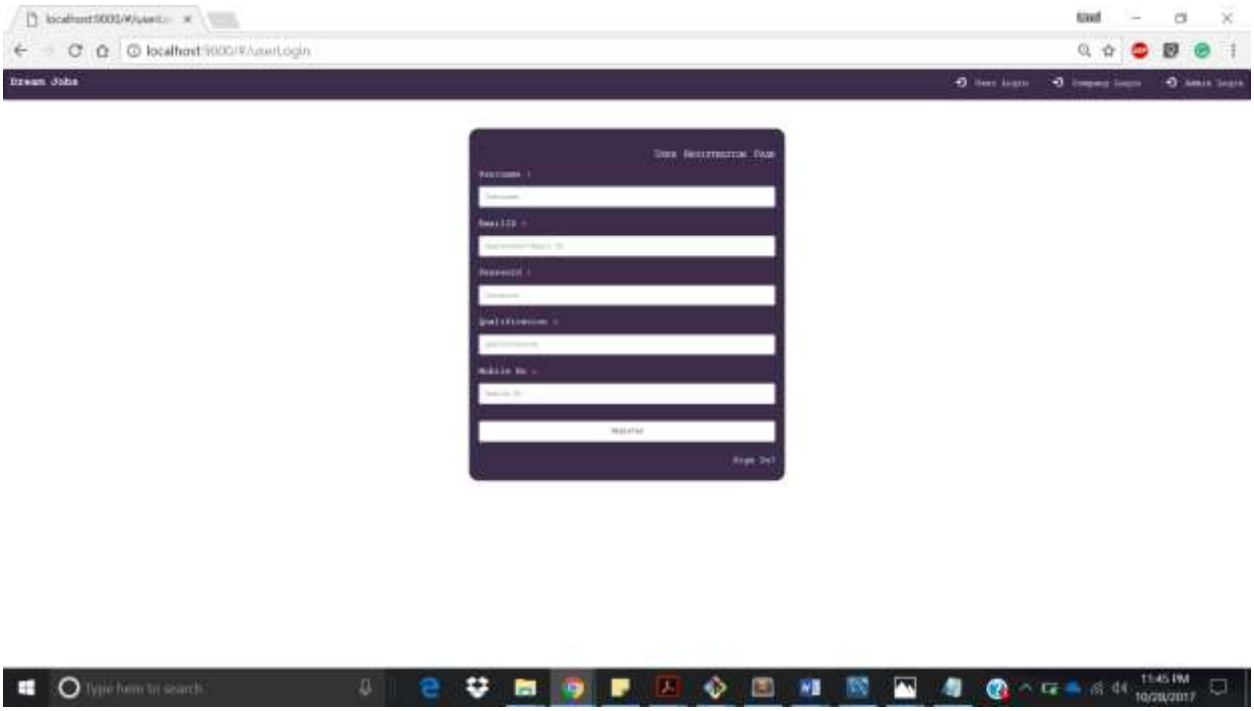


Figure 8.9 Jobseeker Profile Registration Page

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.

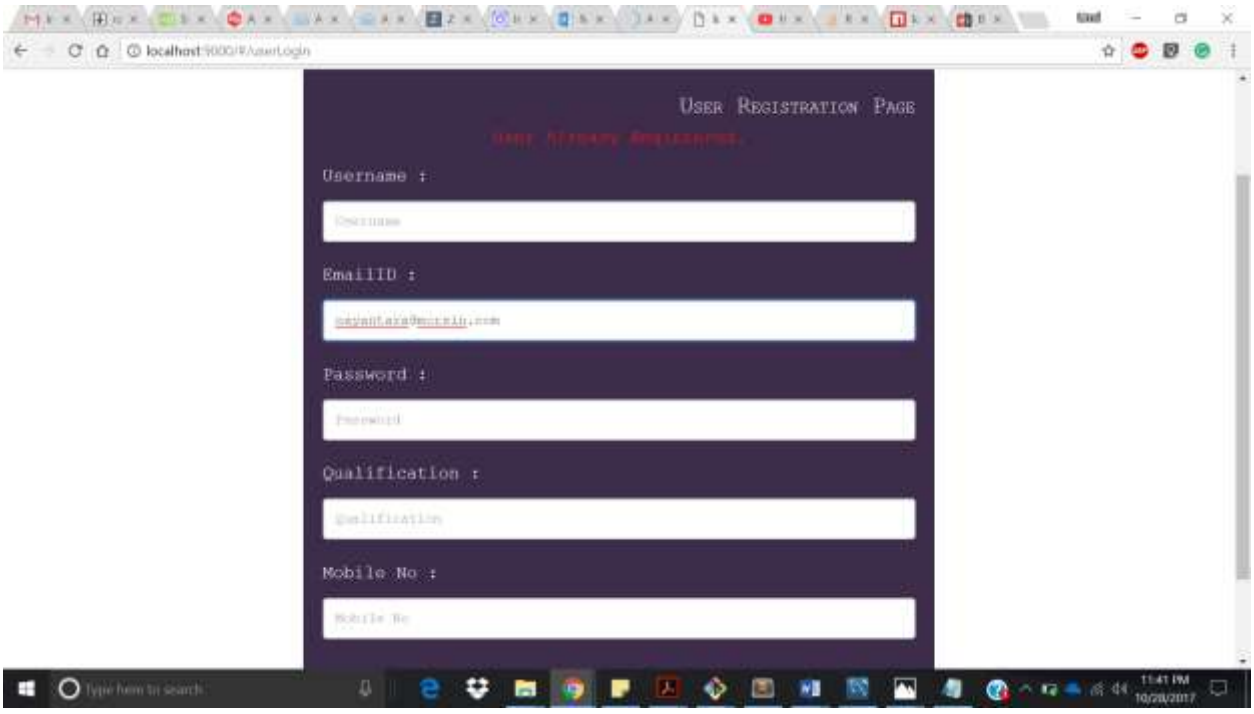


Figure 8.10 Jobseeker Profile Registration Error

Validations are implemented for this web application. Once an email id is registered, the same id cannot be used for registration. The field is unique and cannot have more than one values. If such a scenario happens then an “User already registered” error message is thrown. Finally, when the candidate clicks the sign out button he is again redirected to the first page as shown in figure 8.1 i.e. the user successfully signs out of the portal.

8.2 Admin Module

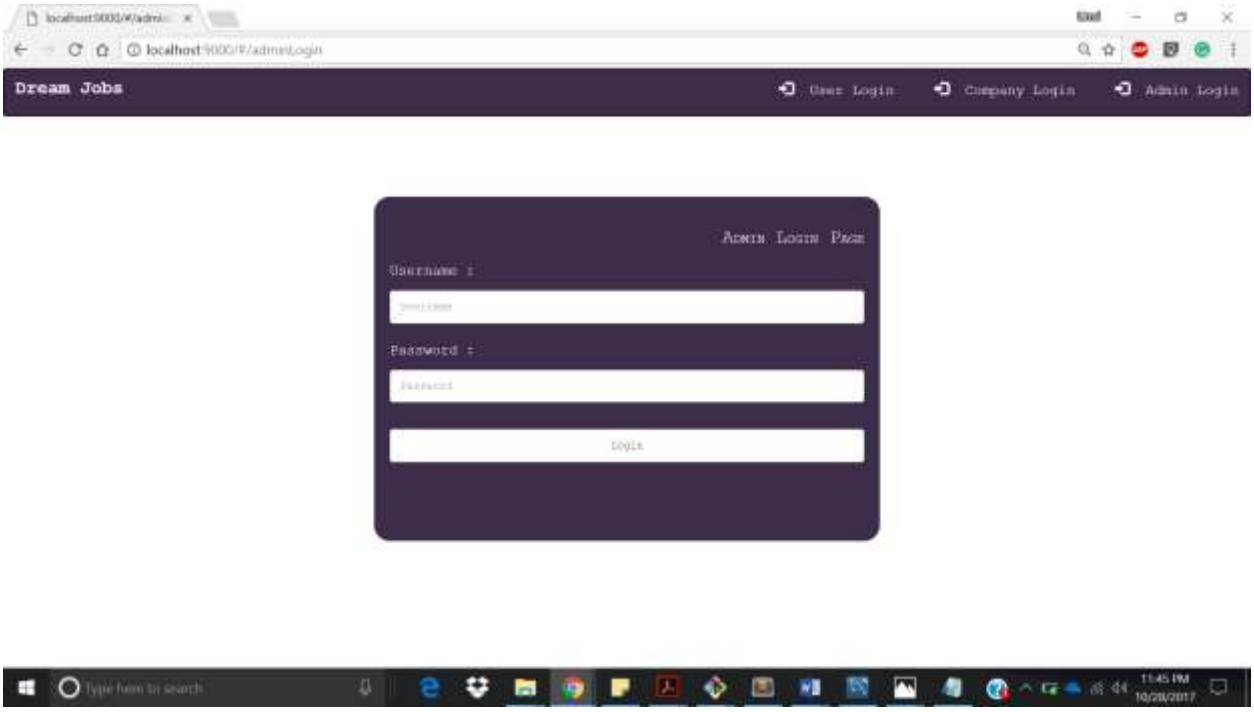


Figure 8.11 Admin Login Page

The admin has a special username and password through which the admin logs in the portal.

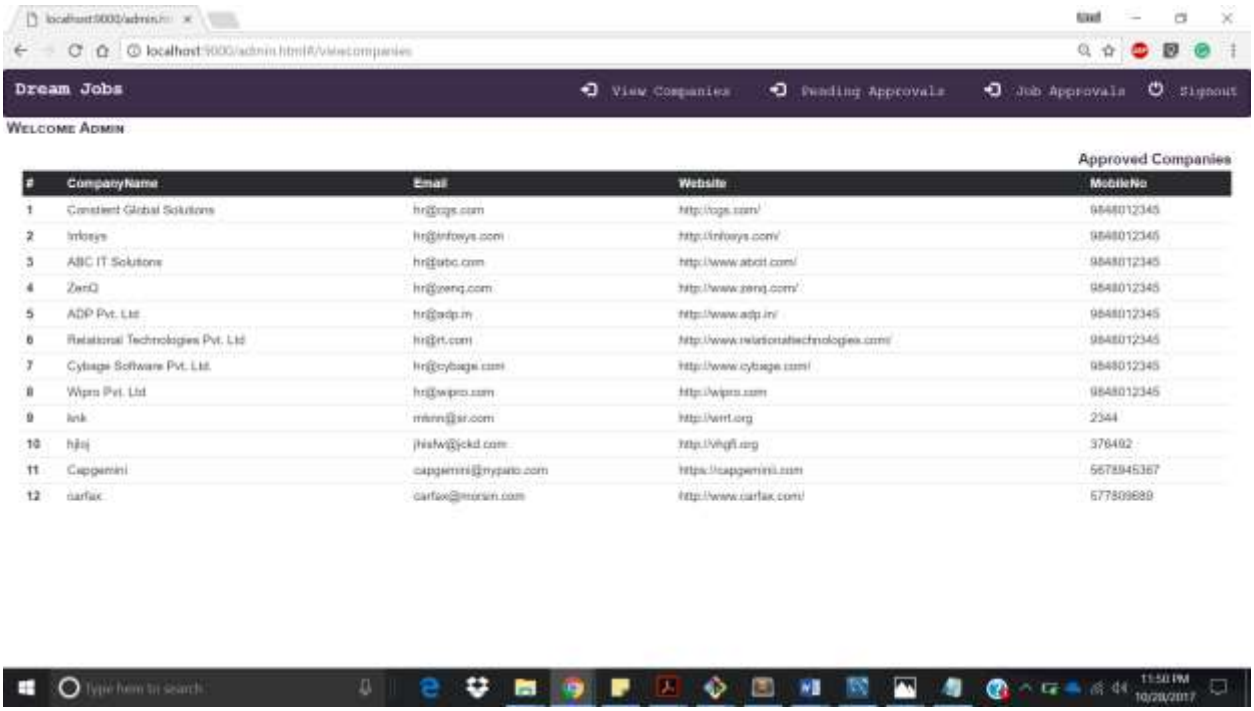


Figure 8.12 Admin Home Page

The admin homepage shows a list of all the companies that are approved to be registered to the job search portal by the admin.

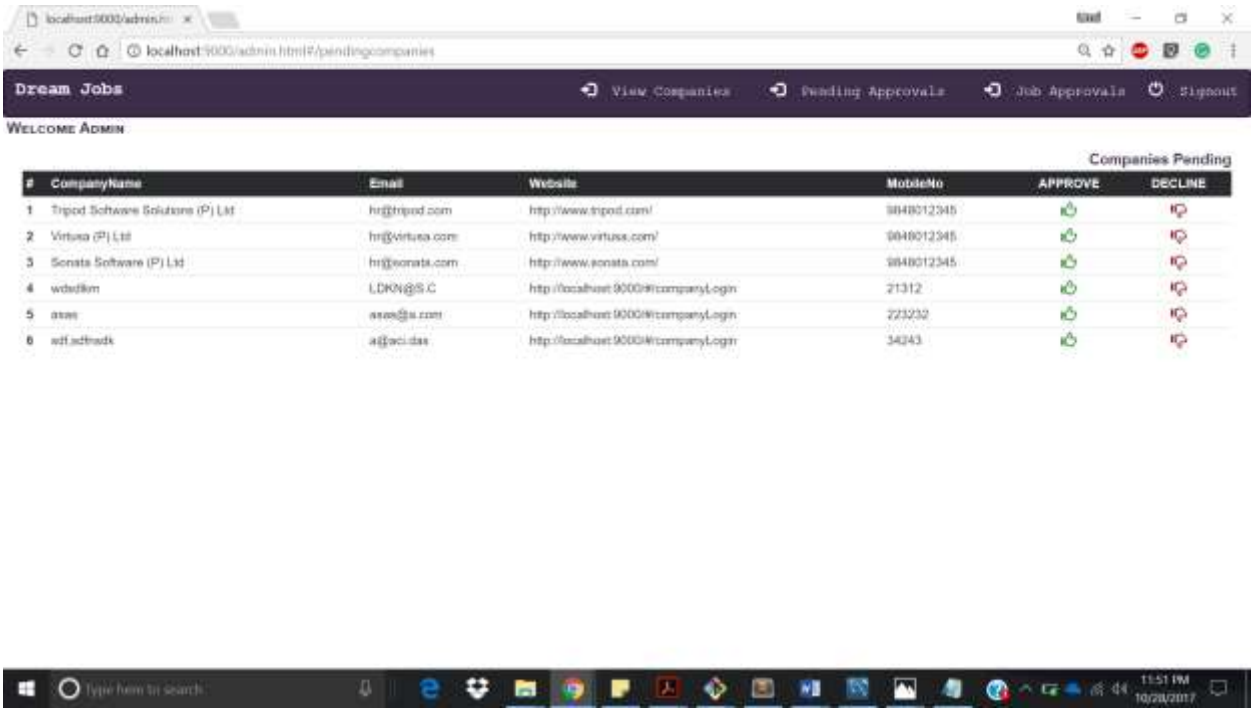


Figure 8.13 Admin View Companies Page

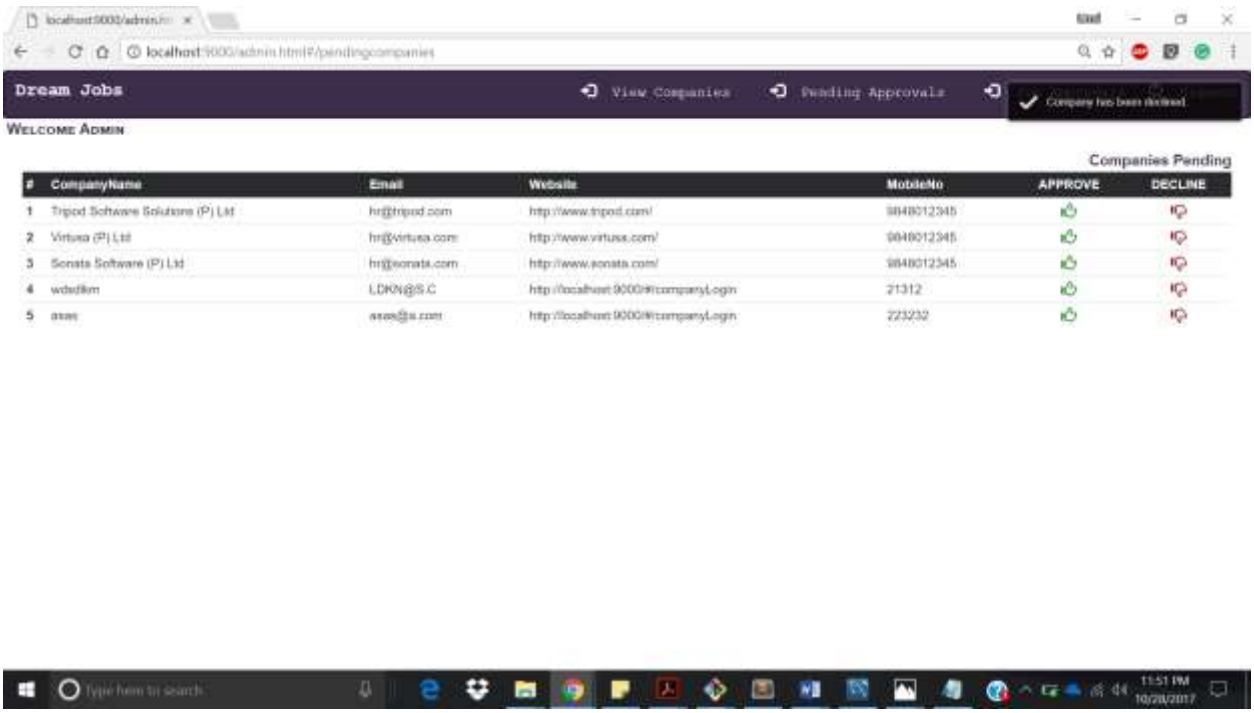


Figure 8.14 Admin Approve/Disapprove Companies Message

Figure 8.13 shows a list of companies that the admin needs to approve/disapprove so that the companies can get registered to the portal. Figure 8.14 shows the feedback message at the right hand top corner when the admin approves a company. Similar message pops up when the admin disapproves a company. Once approved the company is visible in the company list to the admin and the company can login with its credentials to post jobs and find the right candidates.

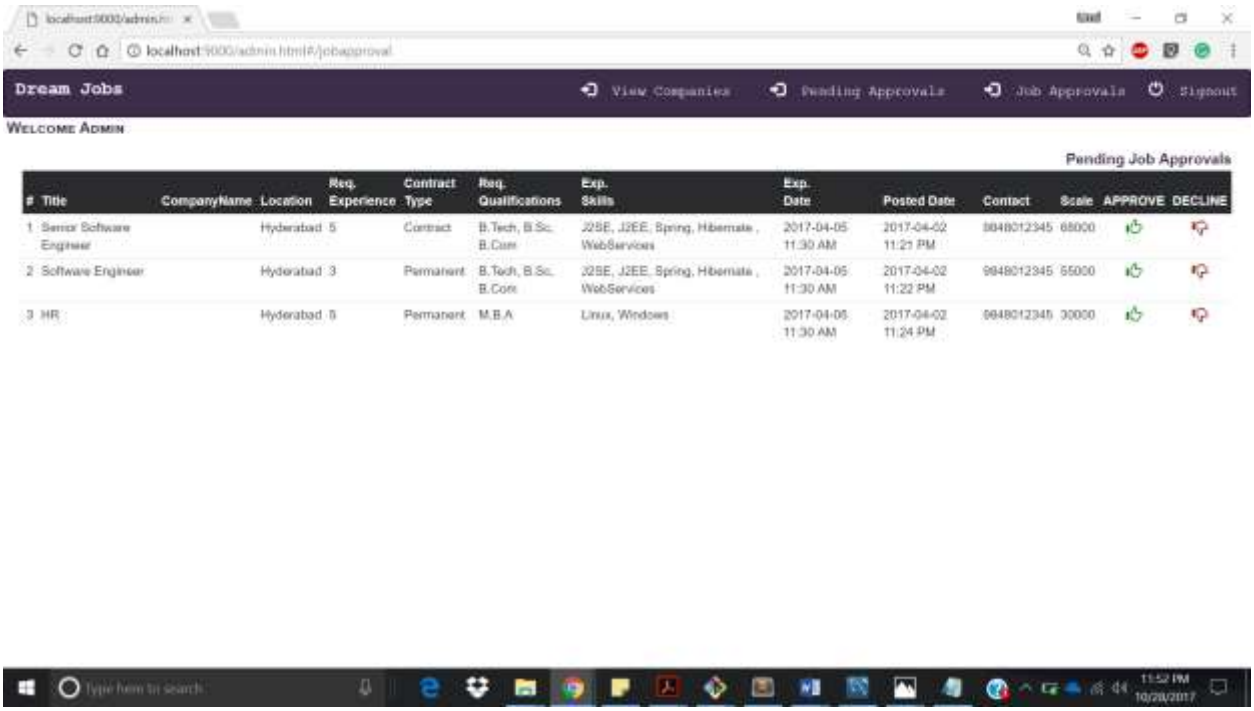


Figure 8.15 Admin Job Approvals Page

Figure 8.15 shows the job required to be approved by the admin so that the job seeker can view the jobs. Once the admin approves/disapproves a job, the status of the job changes to 1/0 in the database and when the status is 1 i.e. approved the job becomes visible to the user to apply. Once approved/disapproved a message like figure 8.14 is shown on the top right corner as the feedback for approval/disapproval. Finally, as the admin logs out he is directed to the page as shown in figure 8.11.

8.3 Company Module

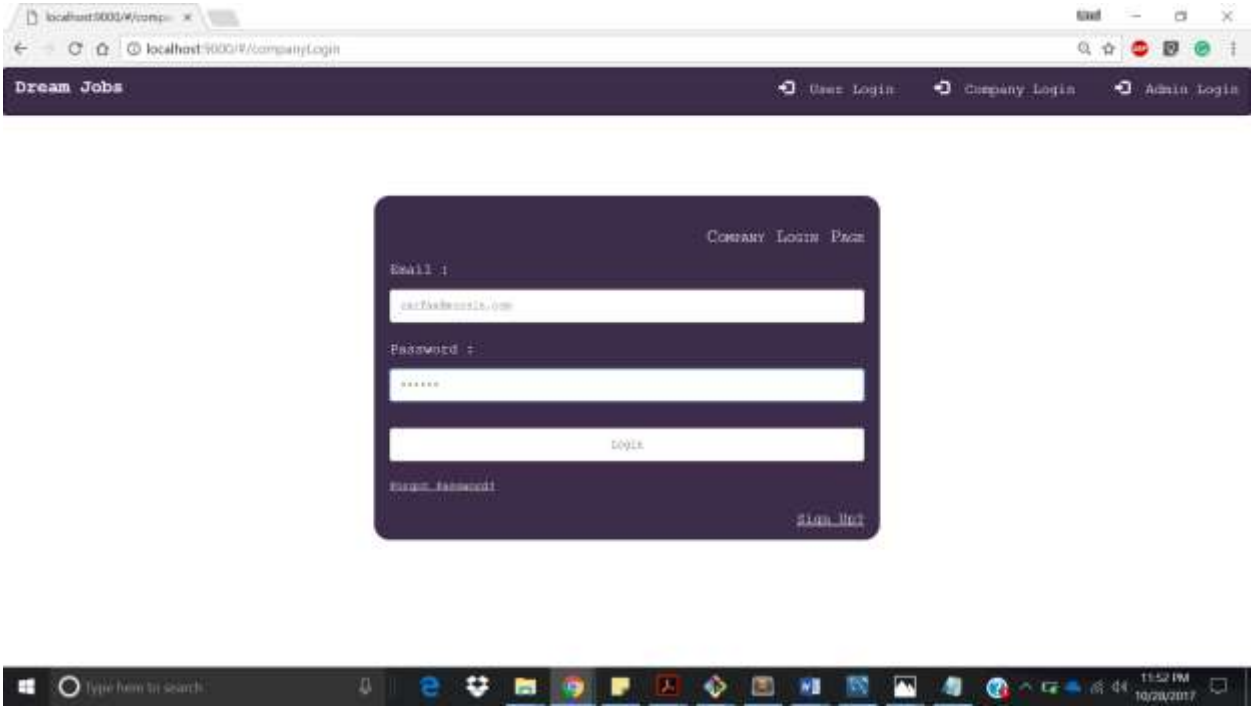


Figure 8.16 Company Login Page

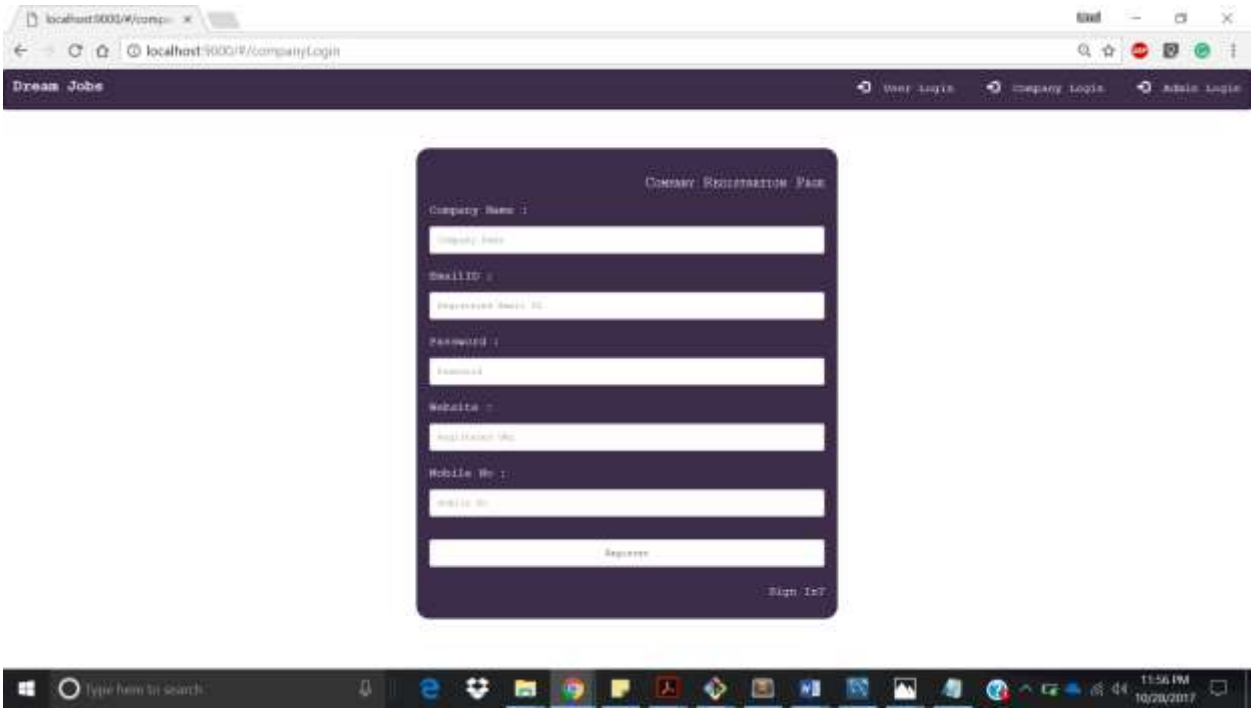


Figure 8.17 Company Registration Page

In figure 8.16 and 8.17 we can see the company login page and also the company registration page. Registered companies once approved by the admin can login and post jobs and view applications. New employers should register first, wait for admin approval/disapproval and then can access the portal.

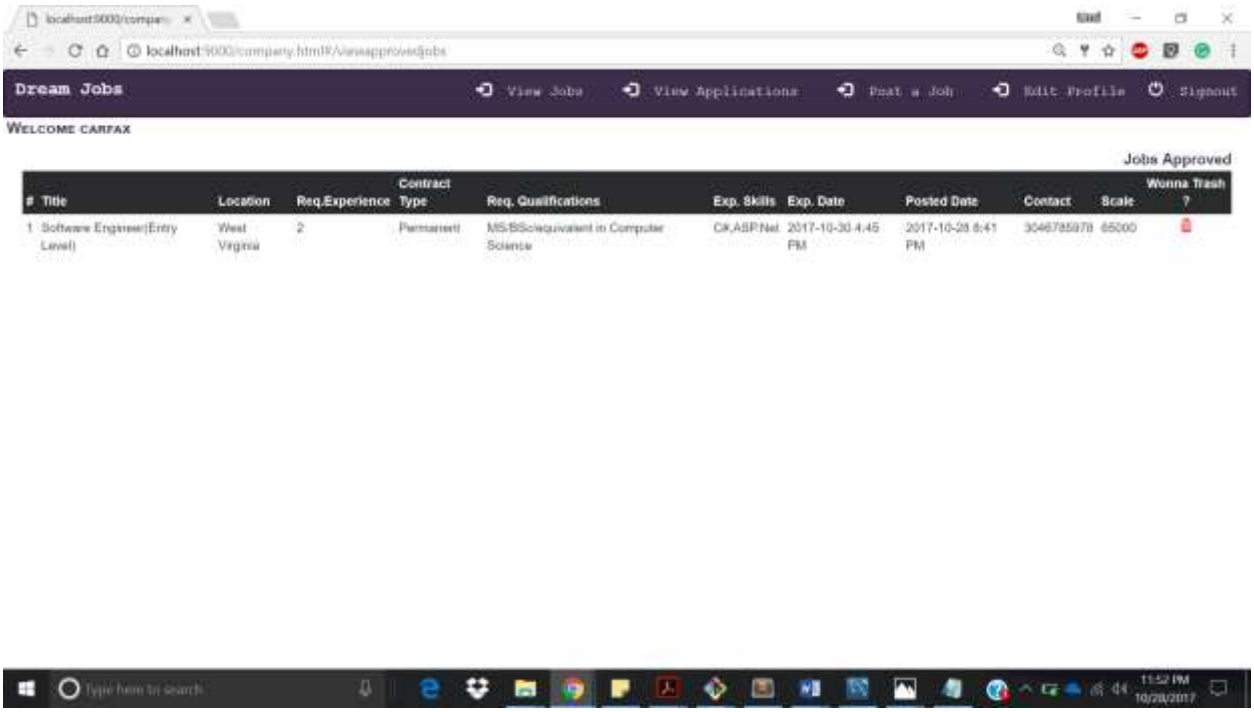


Figure 8.18 Company View Jobs Page

In this page the company can view all the jobs that the company has posted and has been approved by the admin. Once approved the jobs are visible to both the company and the job seeker. The employer can also delete the job if he does not want to keep the job open anymore or if the position is filled.

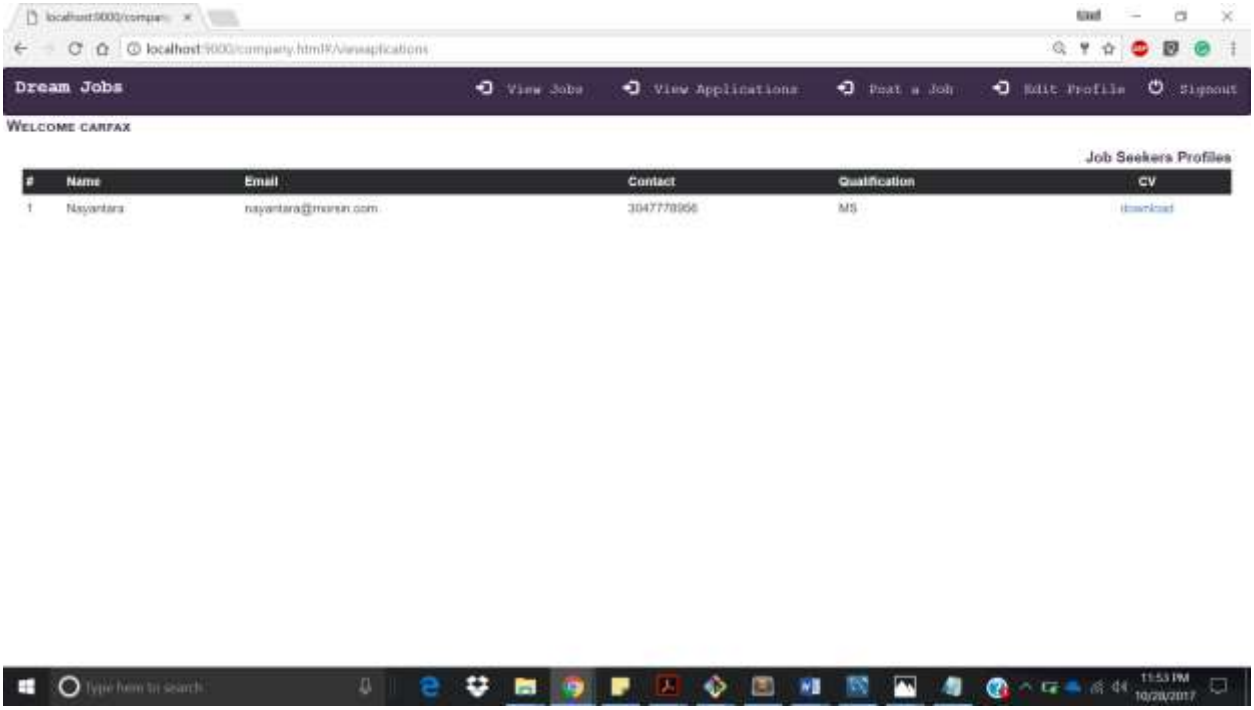


Figure 8.19 Company View Applications Page

The employer can see the list of candidates who have applied for the jobs. They can download and view the resumes of the candidates.

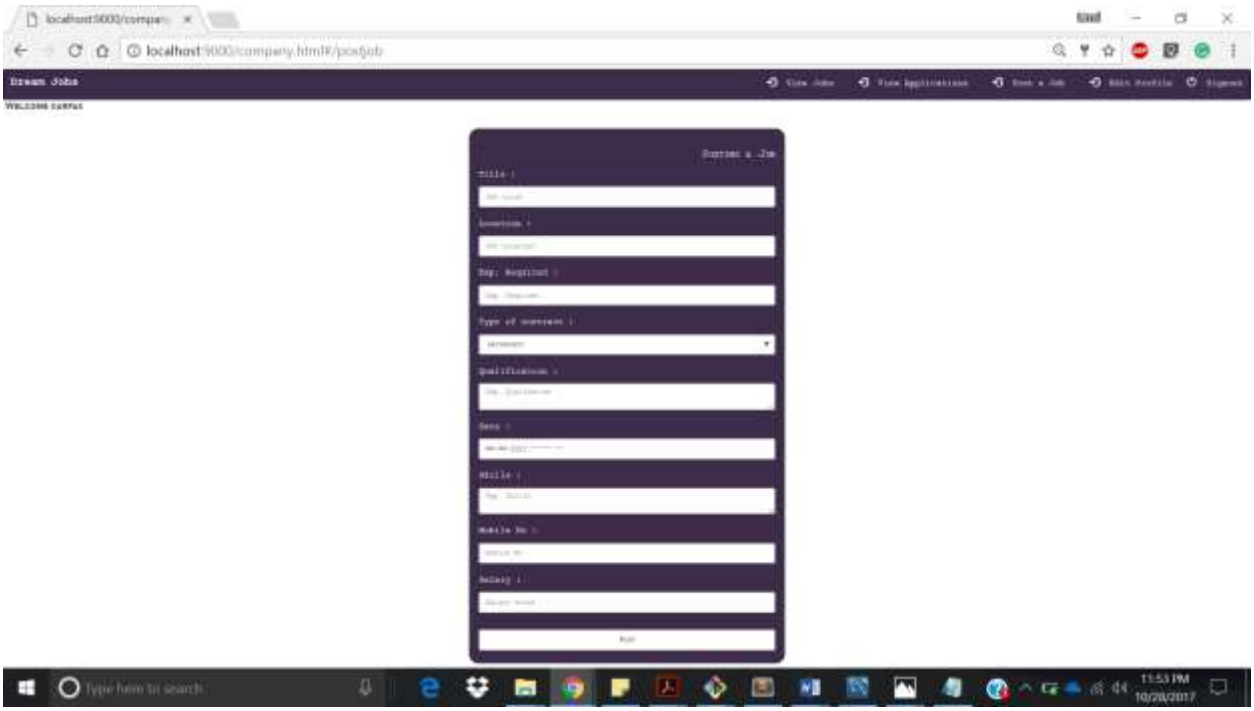


Figure 8.20 Company Post a Job Page

Once the employer wants to post a job, the company fills up this form and submits. The submission is redirected to the admin who would approve/disapprove this job and the company, and the jobseeker can view and apply to this job.

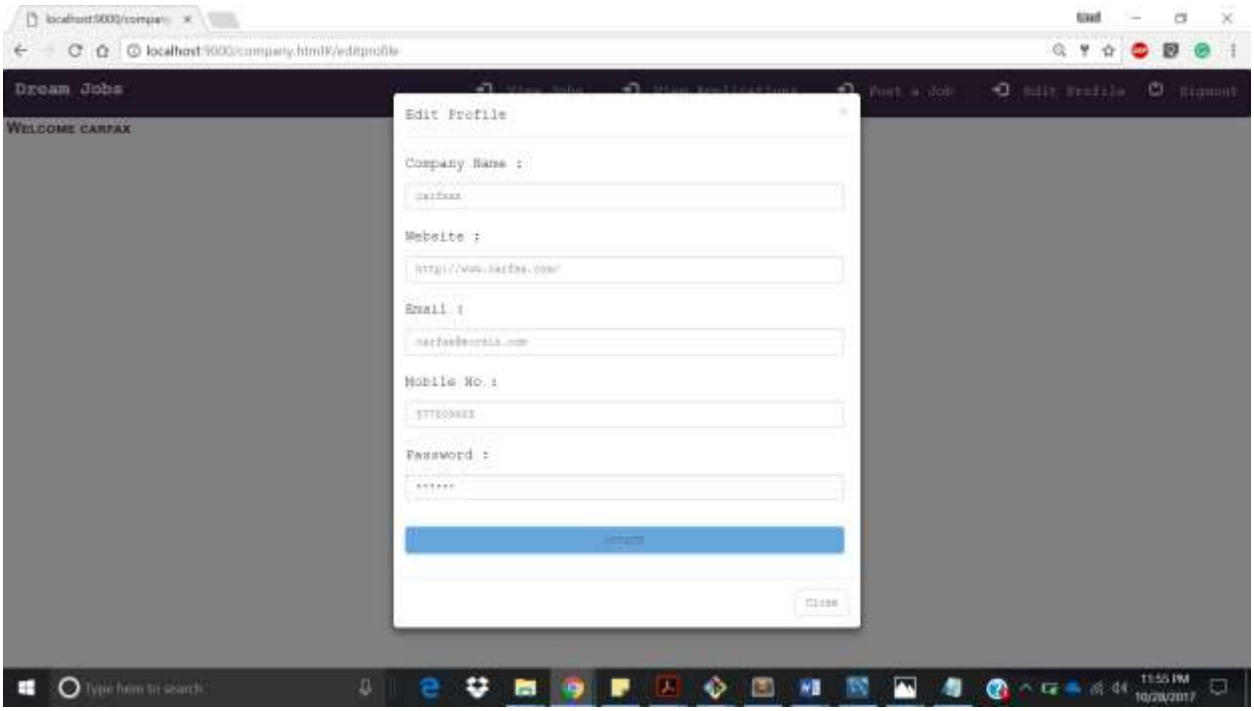


Figure 8.21 Company Edit Profile Page

Once the company wants to edit profile information they have to update the form in figure 8.21 and the change is reflected in the database as well as the view with angular's two-way data binding properties. Finally, if the company signs out he is logged out of the page and is redirected to the first figure shown here.

Chapter 9 - System Metrics

A metric can be thought of like a “parameter” and software system metrics mean whether a software possesses such parameters or meets the requirement of such parameters. Below I would discuss a quantitative metric for my system and a qualitative metric for my system.

9.1 Coupling

Coupling is a qualitative metric of a software system. Coupling means how much some part of the code is dependent on other parts or in other words how much code cannot work independently. The best software systems are developed with the aim to have minimum coupling. My application achieves such a goal, as most part of the code is modular in nature which ensures loose coupling. This means each part of the code can work independently and the degree to which they depend on another part is much low. Changes in one module will not result in a change in another module. These modules can also be reused. Thus, ensuring low coupling, this application ensures easy maintainability and reusability too.

9.2 LOC

SLOC is a quantitative software system metrics. Calculating the number of Lines of Code used gives an idea about how many lines of code has to be maintained. It also can be compared with the number of lines of code that would be approximately required if some other technologies were used. Below is a calculation summary of the LOC that is required by my application:

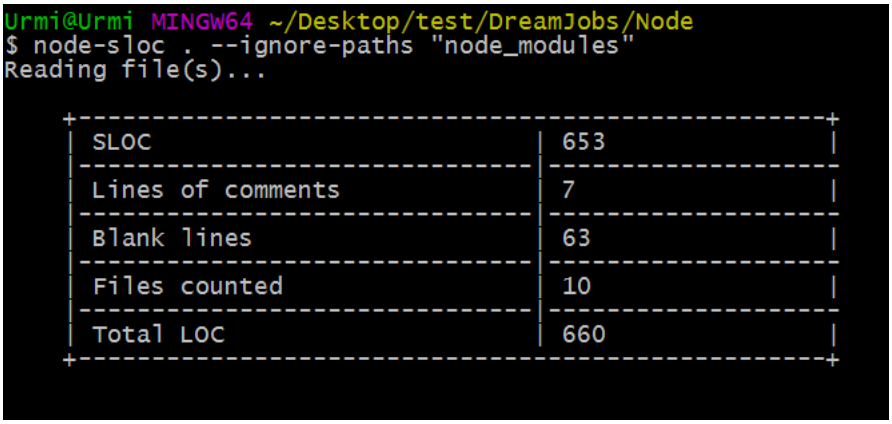


Figure 9.1 Lines of Code on the server side

Figure 9.1 shows the number of lines of Code required to develop the server side which sums up to 660 which includes NodeJS codes.

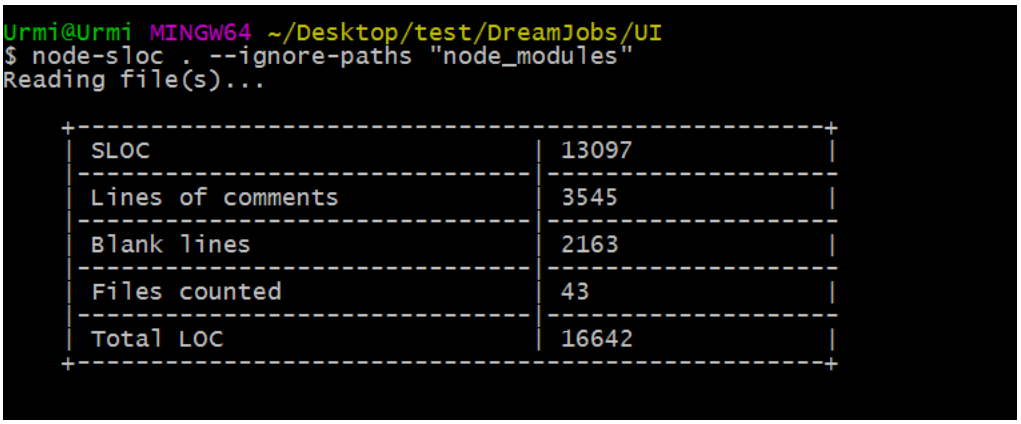


Figure 9.2 Lines of Code on the client side

Figure 9.2 demonstrates the number of lines of code required for developing the client-side application which includes Html, CSS, jQuery and AngularJS.

Approximately, 180 lines of code is required for the Database. Thus, a total count of 17482 Lines of Code is required for my application.

Chapter 10 - 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 NodeJS 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.

10.1 UI Story

After my initial UI construction, I have found that I needed to add some additional capabilities. The primary one was to optimize the search space for which I have developed an advanced search feature. Initially for the advanced search while I was using checkboxes for salary selection, I have then changed it to a slider to optimize space and to also make it more user friendly. Another evolvement of my UI was adding pagination to handle a lot of data. I have tested my application on my friends' computer with different software specifications and found significant differences in load testing. It was on my friend's advice that I have added the purple color to my UI to give it a resemblance to k-state. It was also on a friend's feedback that I have named my application from just a Job Search Portal to Dreams Job.

Chapter 11 - Future Work

There is ample scope of enhancement and adding functionalities to this application. This application can be extended to send automated interview scheduling through acceptance/rejection of Resume. Companies can delete jobs once the job availably period is over automatically. The application can have a job recommendation system based on the frequent search results of different users. The portal can also send email notifications to candidates about certain job availabilities. There can be a feedback or review section for the application. Also unlike the current job description page, user can view job description in a separate page with one click on the job description. The User functionality can be extended to give the user options to save the job and later apply, to upload multiple documents. The application can be more scalable by extending the search functionality based on country, city or area. While this application meets the basic requirements of a job portal eliminating few of the traditional challenges faced like time, money and effort, it can be extended to make the application more dynamic and robust. The User Interface can be made more attractive and user friendly. We can dig through more AngularJS magic capabilities to add additional features to the UI.

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-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-angular-js

[6] Retrieved on 10/10/2017 from https://programmaticponderings.files.wordpress.com/2015/01/mean-stack-architecture-general-third-party-1.png