

Background Verification System

A Project-II Report

Submitted in partial fulfillment of requirement of the

Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

BY

Sumit Porwal

EN18CS301275

Under the Guidance of
Mrs. Priyanka Dhasal



Department of Computer Science & Engineering
Faculty of Engineering
MEDI-CAPS UNIVERSITY, INDORE- 453331

May 2022

Background Verification System

A Project-II Report

Submitted in partial fulfillment of requirement of the

Degree of

BACHELOR OF TECHNOLOGY

in

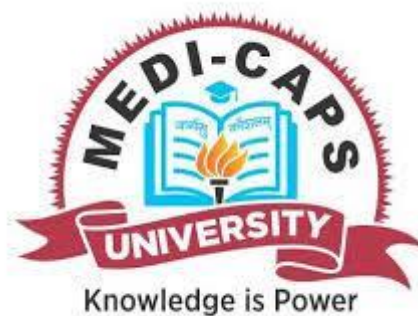
COMPUTER SCIENCE & ENGINEERING

BY

Sumit Porwal

EN18CS301275

Under the Guidance of
Mrs. Priyanka Dhasal



Department of Computer Science & Engineering
Faculty of Engineering
MEDI-CAPS UNIVERSITY, INDORE- 453331

May 2022

Report Approval

The project work “**Background Verification System**” is hereby approved as a creditable study of an engineering/computer application subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approved any statement made, opinion expressed, or conclusion drawn there in; but approve the “Project Report” only for the purpose for which it has been submitted.

Internal Examiner

Name:

Designation

Affiliation

External Examiner

Name:

Designation

Affiliation

Declaration

I/We hereby declare that the project entitled “**Background Verification System**” submitted in partial fulfillment for the award of the degree of Bachelor of Technology in ‘Computer Science and Engineering’ completed under the supervision of **Mrs. Priyanka Dhasal, Assistant Professor, Computer Science and Engineering**, Faculty of Engineering, Medi-Caps University Indore and **Mr. Tarun Pura, Associate Software Developer, Consultadd Services Private Limited** is an authentic work.

Further, I/we declare that the content of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

Sumit Porwal

EN18CS301275

Date:19/05/2022

Certificate

I/We, **Mrs. Priyanka Dhasal** certify that the project entitled “**Background Verification System**” submitted in partial fulfillment for the award of the degree of Bachelor of Technology by **Sumit Porwal** is the record carried out by him/them under my/our guidance and that the work has not formed the basis of award of any other degree elsewhere.



Mrs. Priyanka Dhasal

Computer Science and Engineering

Medi-Caps University, Indore

Mr. Tarun Pura

Engineering

Consultadd Services Private Limited

Dr. Pramod S. Nair

Head of the Department

Computer Science & Engineering

Medi-Caps University, Indore

Offer Letter of the Project work-II/Internship



LETTER OF INTERNSHIP

Name: Sumit porwal

19 December, 2021

Further to your application and subsequent interview with us, we are pleased to offer you the position of **Software Engineer Intern** in our organization.

You will be paid a monthly stipend of **25000/- (Twenty five Thousand only)**.

The location of internship is remote due to COVID 19 after that it will be Pune, and the duration of the Internship is from **Tuesday 4th January, 2022 to Thursday 30th June, 2022**. The work timings will be flexible shift.

Please note, your Internship will be at the discretion of the company which means the company may terminate your internship at any time, based on performance

Please share PAN Card & Aadhar Card at the time of joining.

The internship offer is subject to acceptance and verification of documents.

Terms and conditions of the Internship are given in Annexure one.

With best wishes for a long Association.

For Consultadd Services Pvt. Ltd. Pune

Neha Chiddarwar
HR Associate



www.consultadd.com

Acknowledgements

I would like to express my deepest gratitude to Honorable Chancellor, **Shri R C Mittal**, who has provided me with every facility to successfully carry out this project, and my profound indebtedness to **Prof. (Dr.) Dilip K Patnaik**, Vice Chancellor, Medi-Caps University, whose unfailing support and enthusiasm has always boosted up my morale. I also thank **Prof. (Dr.) D K Panda**, Pro Vice Chancellor, **Dr. Suresh Jain**, Dean Faculty of Engineering, Medi-Caps University, for giving me a chance to work on this project. I would also like to thank my Head of the Department **Dr. Pramod S. Nair** for his continuous encouragement for betterment of the project.

I express my heartfelt gratitude to my **External Guide, Mr. Tarun Pura**, Associate Software Developer, Consultadd Services Pvt. Ltd. as well as to my **Internal Guide, Mrs. Priyanka Dhasal**, Assistant Professor, Department of Computer Science and Engineering, MU, without whose continuous help and support, this project would ever have reached to the completion.

It is their help and support, due to which we became able to complete the design and technical report. Without their support this report would not have been possible.

Sumit Porwal

B.Tech. IV Year

Department of Computer Science & Engineering

Faculty of Engineering

Medi-Caps University, Indore

Abstract

As a part of my under-graduate program, I joined ConsultAdd Services Pvt. Ltd. on 4th January 2022 as a Software Engineer Intern. In the initial training of 2 months, I went through several internal trainings and evaluations related to different technologies and software development. After our training I am part of the development team for the project named Background Verification System.

ConsultAdd is a niche IT Training | Recruiting | Consulting solution provider for global clients in diverse vertical and horizontal streams such as finance and accounts, IT, health care, financial services, human resource, and knowledge process outsourcing and staffing.

ConsultAdd Information Technology Solutions has an unmatched ability to help its client's businesses grow by improving their back-office processes, while significantly reducing costs.

Table of Contents

		Page No.
	Report Approval	iii
	Declaration	iv
	Certificate	v
	Offer Letter of the Project work-II/Internship	vi
	Acknowledgement	vii
	Abstract	viii
	Table of Contents	ix
	List of Figures	xi
	Abbreviations	xii
Chapter 1	Introduction	
	1.1 Introduction	1
	1.2 Organization Profile	1
	1.3 Job Responsibilities	2
	1.4 Technologies Learned During Training	2
Chapter 2	Version Control System	
	2.1 Version Control System - Introduction	3
	2.2 Git	3
	2.3 Github	5
Chapter 3	Java	
	3.1 Introduction to Java	6
	3.2 Spring Framework	7
	3.3 Spring Boot	8
	3.4 Testing in Spring Boot	9
Chapter 4	Database Management System	
	4.1 DBMS	11
	4.2 SQL	11
	4.3 MySQL	12
	4.4 MongoDB	13
Chapter 5	Frontend Technologies	
	5.1 ReactJS	14
	5.2 Angular	15
	5.3 Redux	15
Chapter 6	Apache Kafka	
	6.1 Messaging System	17
	6.2 Introduction to Apache Kafka	18
Chapter 7	SDLC	
	7.1 Introduction to Software Development	21
	7.2 Waterfall Method	22

	7.3 Agile Methodology	23
	7.4 Software Development Tools	24
Chapter 8	Project	
	8.1 Introduction	26
	8.2 Problem Statement	26
	8.3 Objectives	26
	8.4 Architecture	27
	8.5 Technologies Used	27
	8.6 Use Case Diagram	28
	8.7 Activity Diagram	28
	8.8 Sequence Diagram	29
Chapter 9	Project Testing	
	9.1 Testing Methods Applied	30
	9.2 Test Cases	30
Chapter 10	Snapshots of Project	33
Chapter 11	Summary and Conclusion	37
Chapter 12	Bibliography	38

List of Figures

	Figures	Page No.
Figure 8.1	Project Architecture	27
Figure 8.2	Use Case Diagram	28
Figure 8.3	Activity Diagram	28
Figure 8.4	Employer Sequence Diagram	29
Figure 8.5	Candidate Sequence Diagram	29

Abbreviations

Abbreviations	Full Form
JVM	Java Virtual Machine
JDK	Java Development Kit
IDE	Integrated Development Environment
SDK	Software Development Kit
DBMS	Database Management System
SQL	Structured Query Language
SDLC	Software Development Lifecycle
JS	JavaScript
API	Application Programming Interface

Chapter-1

Introduction

1.1 Introduction

The Project “Background Verification System” is a web application based on the Background Verification of the documents of the newly hired employees. The old application already exists which is monolithic application. It has been migrated to a MicroService based application. It is based on event driven architecture. It was mainly built using Spring Boot. It was tested using Junit and Mockito. It is using gradle as the build tool. It is using gitlab for git versioning and also uses argoCD for continuous deployment.

1.2 Organization Profile

ConsultAdd Inc is a leading provider of Information Technology consulting and business process services. We offer strategic insights, technological expertise, and industry experience. ConsultAdd is one of the fastest growing IT consulting companies specializing in finance, banking, insurance, healthcare, retail, e-commerce domains and enterprise web development, data warehouse, business intelligence, big data in technologies.

With over 400+ professionals working on information technology projects across united states, ConsultAdd is headquartered in TX and has the presence in Virginia, Pennsylvania, New York. ConsultAdd's offshore center is based in India at Pune.

Website-

<https://consultadd.com/>

Company size-

400-500 employees

Headquarters-

Texas, United States

Pune, India (Offshore Center)

Founded-

2011

Specialties-

IT Consulting, Product Development, IT Training, Amazon Web Services (AWS), Python, Java, Software Development, Cloud Development, DevOps, Elastic Stack (ELK), People First, Culture First, JavaScript, and React JS

Major Clients-

Google, Cisco, Apple, Lululemon, Kasasa, SynapseFi, Fannie-mae and many more.

1.3 Job Responsibilities

My profile will be Software Engineer Intern. I have been contributing in:

- Rest API development with Spring Boot.
- Developing Producer and Consumer API's for producing and consuming messages from/to kafka topics.
- Performing Unit-testing on the project.
- Integrating with tools like Sonarqube and Jacoco for checking the code coverage.
- Integrating Swagger for documenting the API's.

1.4 Technologies Learned During Training

The technologies learned by me during the training period are:

- HTML5, CSS3, JavaScript
- Java
- Spring Boot
- Junit and Mockito
- REST API's
- Microservices
- Database - MongoDB, MySQL
- Git & GitHub
- Postman
- Docker
- ReactJS
- Redux
- Angular

Chapter-2

Version Control System

2.1 Version Control System – Introduction

Version control systems are a class of software tools that help a software team manage changes to source code over the period of time. Version control software keeps track of every modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistakes while minimizing disruption to all team members.

Benefits of version control systems:

Developing software without using version control is risky, like not having backups. Version control can also enable developers to move faster and it allows software teams to preserve efficiency and agility as the team scales to include more developers. The primary benefits you should expect from version control are as follows.

- A complete long-term change history of every file.
- Branching and merging.
- Traceability.

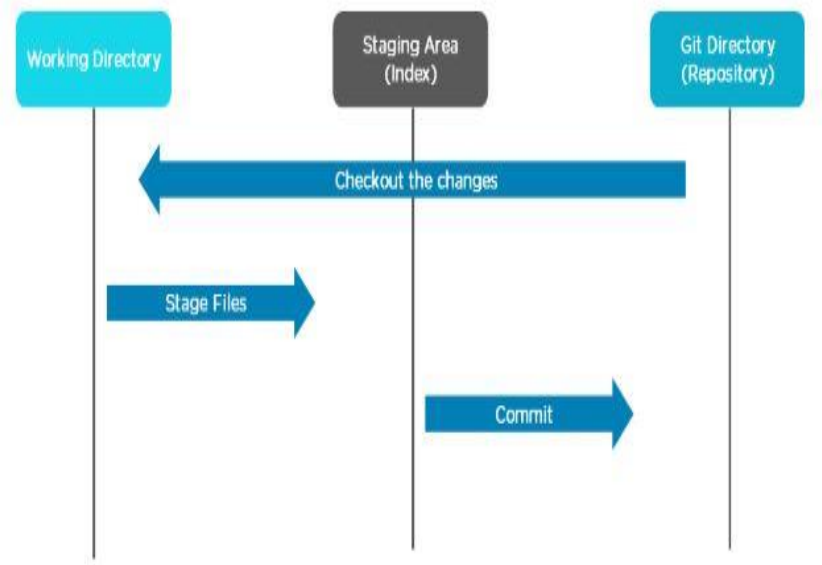
Version Control Systems (VCS) have seen great improvements over the past few decades and some are better than others. VCS are sometimes known as SCM (Source Code Management) tools or RCS (Revision Control System)

2.2 Git

Git is a mature, actively maintained open-source project originally developed in 2005 by Linus Torvalds, the famous creator of the Linux operating system kernel. A staggering number of software projects rely on Git for version control, including commercial projects as well as open-source. Developers who have worked with Git are well represented in the pool of available software development talent and it works well on a wide range of operating systems and IDEs (Integrated Development Environments).

Features of Git

- Tracks history
- Free and open source
- Supports non-linear development
- Creates backups
- Scalable
- Supports collaboration
- Branching is easier
- Distributed development



Useful Commands

- `git init`
- `git clone`
- `git add`
- `git commit`
- `git status`
- `git push`
- `git pull`
- `git diff`
- `git log`
- `git reset`
- `git checkout`
- `git merge`

2.3 Github

Git Hub is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management (SCM) functionality of Git, plus its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, continuously integration and wikis for every project.

Other than GitHub we also have AWS Code Commit, GitLab, Bitbucket, etc.

GitHub

 **Bitbucket**



GitLab



Azure DevOps

Chapter-3

Java



3.1 Introduction to Java

Java is a programming language and computing platform first released by Sun Microsystems in 1995. There are lots of applications and websites that will not work unless you have Java installed, and more are created every day. Java is fast, secure, and reliable. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

Java platform is a collection of programs that help to develop and run programs written in the Java programming language. Java platform includes an execution engine, a compiler, and a set of libraries. JAVA is platform-independent language. It is not specific to any processor or operating system.

Java is a programming language created by James Gosling from Sun Microsystems (Sun) in 1991. The target of Java is to write a program once and then run this program on multiple operating systems. The first publicly available version of Java (Java 1.0) was released in 1995. Sun Microsystems was acquired by the Oracle Corporation in 2010. Oracle has now the steering ship for Java. Oracle continues this project called OpenJDK.

Features of Java

- Simple
- Object-Oriented
- Portable
- Platform independent
- Secured
- Robust
- Architecture neutral
- Interpreted
- High Performance

- Multithreaded
- Distributed
- Dynamic

3.2 Spring Framework



Spring is a *lightweight* framework. It can be thought of as a *framework of frameworks* because it provides support to various frameworks such as Struts, Hibernate, Tapestry, EJB, JSF, etc. The framework, in broader sense, can be defined as a structure where we find solution of the various technical problems.

IoC Container

The IoC container is responsible to instantiate, configure and assemble the objects. The IoC container gets informations from the XML file and works accordingly. The main tasks performed by IoC container are:

- to instantiate the application class
- to configure the object
- to assemble the dependencies between the objects

Dependency Injection

Dependency Injection (DI) is a design pattern that removes the dependency from the programming code so that it can be easy to manage and test the application. Dependency Injection makes our programming code loosely coupled.

Spring framework provides two ways to inject dependency

- By Constructor
- By Setter method

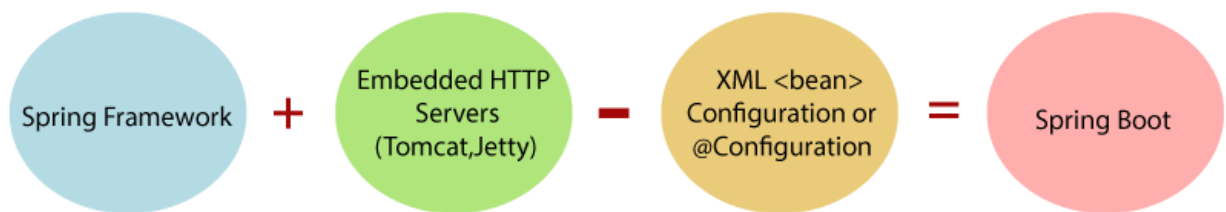
Autowiring

Autowiring feature of the spring framework enables you to inject the object dependency implicitly. It internally uses setter or constructor injection. It works with reference only.

3.3 Spring Boot

Spring Boot is a project that is built on the top of the Spring Framework. It provides an easier and faster way to set up, configure, and run both simple and web-based applications.

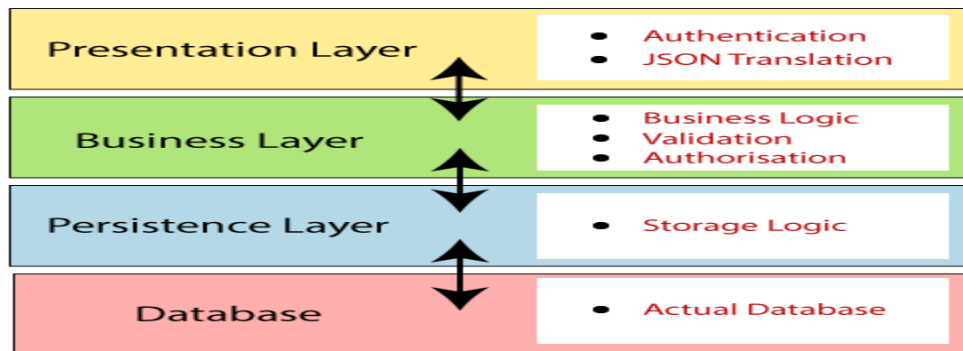
It is a Spring module that provides the **RAD (*Rapid Application Development*)** feature to the Spring Framework. It is used to create a stand-alone Spring-based application that you can just run because it needs minimal Spring configuration.



Spring Boot Features

- Web Development
- Spring Application
- Application events and listeners
- Admin features
- Externalized Configuration
- Properties Files
- YAML Support
- Type-safe Configuration
- Logging
- Security

Spring Boot Architecture



Spring Boot is a module of the Spring Framework. It is used to create stand-alone, production-grade Spring Based Applications with minimum efforts. It is developed on top of the core Spring Framework.

Spring Boot follows a layered architecture in which each layer communicates with the layer directly below or above (hierarchical structure) it.

Before understanding the **Spring Boot Architecture**, we must know the different layers and classes present in it. There are **four** layers in Spring Boot are as follows:

- **Presentation Layer**
- **Business Layer**
- **Persistence Layer**
- **Database Layer**

3.4 Testing in Spring Boot

Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance.

Unit Testing is a one of the testing done by the developers to make sure individual unit or component functionalities are working fine.

Tools used for testing in Spring Boot

- **JUnit:** JUnit is a unit testing framework for Java programming language. JUnit has been important in the development of test-driven development, and is one of a family of unit testing frameworks collectively known as xUnit, that originated with JUnit. JUnit promotes the idea of "first testing then coding", which emphasizes on setting up the test data for a piece of code that can be tested first and then implemented. This approach is like "test a little, code a little, test a little, code a little." It increases the productivity of the programmer and the stability of program code, which in turn reduces the stress on the programmer and the time spent on debugging.
- **Mockito:** Mockito is a mocking framework, JAVA-based library that is used for effective unit testing of JAVA applications. Mockito is used to mock interfaces so that a dummy functionality can be added to a mock interface that can be used in unit testing. It makes unit testing highly effective with clean tests, thanks to dependency injection and compile-time checks.
- **SonarQube:** SonarQube is an open-source and standalone service that provides an overview of the overall health of our source code by measuring code quality and code coverage. SonarQube inspects and evaluates everything that affects our codebase, from minor styling details to critical design errors.
- **Jacoco:** *Jacoco* is an open source project, which can be used to check production code for test code coverage. It creates reports and integrates well with IDEs like the Eclipse IDE. Integration is also available for other IDEs and continuous integration environments. So there are also Gradle, SonarQube and Jenkins plugins to make these code coverage checks outside the IDE and therefore globally available to the development team.

Chapter-4

Database Management System

4.1 DBMS

What is a Database?

A database is a collection of related data which represents some aspect of the real world. A database system is designed to be built and populated with data for a certain task.

What is DBMS?

Database Management System (DBMS) is a software for storing and retrieving users' data while considering appropriate security measures. It consists of a group of programs which manipulate the database. The DBMS accepts the request for data from an application and instructs the operating system to provide the specific data. In large systems, a DBMS helps users and other third-party software to store and retrieve data. DBMS allows users to create their own databases as per their requirement. The term “DBMS” includes the user of the database and other application programs. It provides an interface between the data and the software application.

Characteristics of Database Management System

- Provides security and removes redundancy
- Self-describing nature of a database system
- Insulation between programs and data abstraction
- Support of multiple views of the data
- Sharing of data and multiuser transaction processing
- DBMS allows entities and relations among them to form tables.
- It follows the ACID concept (Atomicity, Consistency, Isolation, and Durability).

4.2 SQL

SQL (Structured Query Language) is used to perform operations on the records stored in the database such as updating records, deleting records, creating and modifying tables, views, etc.

SQL is just a query language; it is not a database. To perform SQL queries, you need to install any database, for example, Oracle, MySQL, MongoDB, PostgreSQL, SQL Server, DB2, etc.

What is SQL?

- SQL stands for Structured Query Language.
- It is designed for managing data in a relational database management system (RDBMS).
- It is pronounced as S-Q-L or sometime See-Qwell.
- SQL is a database language, it is used for database creation, deletion, fetching rows, and modifying rows, etc.
- SQL is based on relational algebra and tuple relational calculus.

All DBMS like MySQL, Oracle, MS Access, Sybase, Informix, PostgreSQL, and SQL Server use SQL as standard database language.

4.3 MySQL

MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. It is supported by Oracle Company.



Our MySQL tutorial includes all topics of MySQL database that provides for how to manage database and to manipulate data with the help of various SQL queries. These queries are: insert records, update records, delete records, select records, create tables, drop tables, etc. There are also given MySQL interview questions to help you better understand the MySQL database.

MySQL is a Relational Database Management System (RDBMS) software that provides many things, which are as follows:

- It defines the database relationship in the form of tables (collection of rows and columns), also known as relations.
- It provides the Referential Integrity between rows or columns of various tables.
- It allows us to update the table indexes automatically.
- It uses many SQL queries and combines useful information from multiple tables for the end-users.

4.4 MongoDB

MongoDB is a cross-platform, document oriented database that provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document.



Database

Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases.

Collection

Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database. Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection are of similar or related purpose.

Document

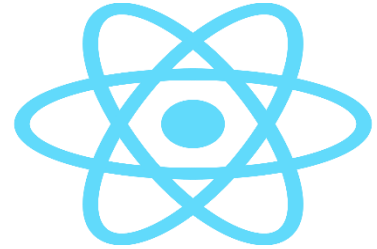
A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.

Chapter-5

Frontend Technologies

5.1 ReactJS

ReactJS is a **declarative, efficient**, and flexible **JavaScript library** for building reusable UI components. It is an open-source, component-based front end library which is responsible only for the view layer of the application. It was initially developed and maintained by Facebook and later used in its products like WhatsApp & Instagram.



Why we use ReactJS?

The main objective of ReactJS is to develop User Interfaces (UI) that improves the speed of the apps. It uses virtual DOM (JavaScript object), which improves the performance of the app. The JavaScript virtual DOM is faster than the regular DOM. We can use ReactJS on the client and server-side as well as with other frameworks. It uses component and data patterns that improve readability and helps to maintain larger apps.

ReactJS Features

Currently, ReactJS gaining quick popularity as the best JavaScript framework among web developers. It is playing an essential role in the front-end ecosystem. The important features of ReactJS are as following.

- JSX
- Components
- One-way Data Binding
- Virtual DOM
- Simplicity
- Performance

5.2 Angular

Angular is a JavaScript framework written in TypeScript and maintained by Google. It enables users to develop and test large applications easily. It has surpassed Javascript for developing single-page applications that require modularity, testability, and developer productivity.



Features of Angular

- **Declarative UI**

Angular uses HTML to define the UI of an application. As HTML is a declarative language, you do not have to define the flows of the program. You can simply define what you require and Angular will take care of it.

- **Typescript**

Angular applications use TypeScript language, which provides higher security as it supports types (primitives, interfaces, etc.) and helps eliminate errors early when writing the code.

- **Easy Testing**

Testing is extremely simple in Angular. The modules have their own application parts, which enable you to perform automated testing. It follows one file, one module principle where you don't even need to remember the module loading order.

5.3 Redux

Redux is a predictable state container for JavaScript apps. As the application grows, it becomes difficult to keep it organized and maintain data flow. Redux solves this problem by managing application's state with a single global object called Store. Redux fundamental principles help in maintaining consistency throughout your application, which makes debugging and testing easier.



More importantly, it gives you live code editing combined with a time-travelling debugger. It is flexible to go with any view layer such as React, Angular, Vue, etc.

Principles of Redux

Predictability of Redux is determined by three most important principles as given below –

- **Single Source of Truth**

The state of your whole application is stored in an object tree within a single store. As whole application state is stored in a single tree, it makes debugging easy, and development faster.

- **State is Read-only**

The only way to change the state is to emit an action, an object describing what happened. This means nobody can directly change the state of your application.

- **Changes are made with pure functions**

To specify how the state tree is transformed by actions, you write pure reducers. A reducer is a central place where state modification takes place. Reducer is a function which takes state and action as arguments, and returns a newly updated state.

Chapter-6

Apache Kafka

6.1 Messaging System

A Messaging System is responsible for transferring data from one application to another, so the applications can focus on data, but not worry about how to share it. Distributed messaging is based on the concept of reliable message queuing. Messages are queued asynchronously between client applications and messaging system. Two types of messaging patterns are available – one is point to point and the other is publish-subscribe (pub-sub) messaging system. Most of the messaging patterns follow **pub-sub**.

Point to Point Messaging System

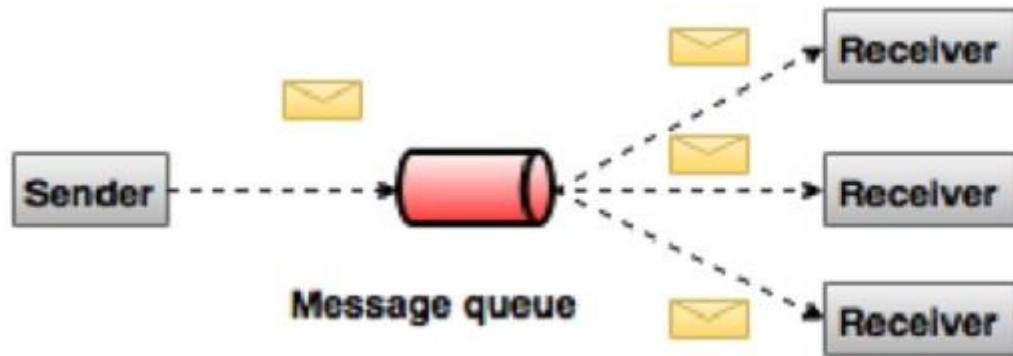
In a point-to-point system, messages are persisted in a queue. One or more consumers can consume the messages in the queue, but a particular message can be consumed by a maximum of one consumer only. Once a consumer reads a message in the queue, it disappears from that queue. The typical example of this system is an Order Processing System, where each order will be processed by one Order Processor, but Multiple Order Processors can work as well at the same time. The following diagram depicts the structure.



Publish-Subscribe Messaging System

In the publish-subscribe system, messages are persisted in a topic. Unlike point-to-point system, consumers can subscribe to one or more topic and consume all the messages in that topic. In the Publish-Subscribe system, message producers are called publishers and message

consumers are called subscribers. A real-life example is Dish TV, which publishes different channels like sports, movies, music, etc., and anyone can subscribe to their own set of channels and get them whenever their subscribed channels are available.



6.2 Introduction to Apache Kafka

Apache Kafka is a distributed data store optimized for ingesting and processing streaming data in real-time. Streaming data is data that is continuously generated by thousands of data sources, which typically send the data records in simultaneously. A streaming platform needs to handle this constant influx of data, and process the data sequentially and incrementally.



Kafka provides three main functions to its users:

- Publish and subscribe to streams of records
- Effectively store streams of records in the order in which records were generated
- Process streams of records in real time

Kafka is primarily used to build real-time streaming data pipelines and applications that adapt to the data streams. It combines messaging, storage, and stream processing to allow storage and analysis of both historical and real-time data.

Why would you use Kafka?

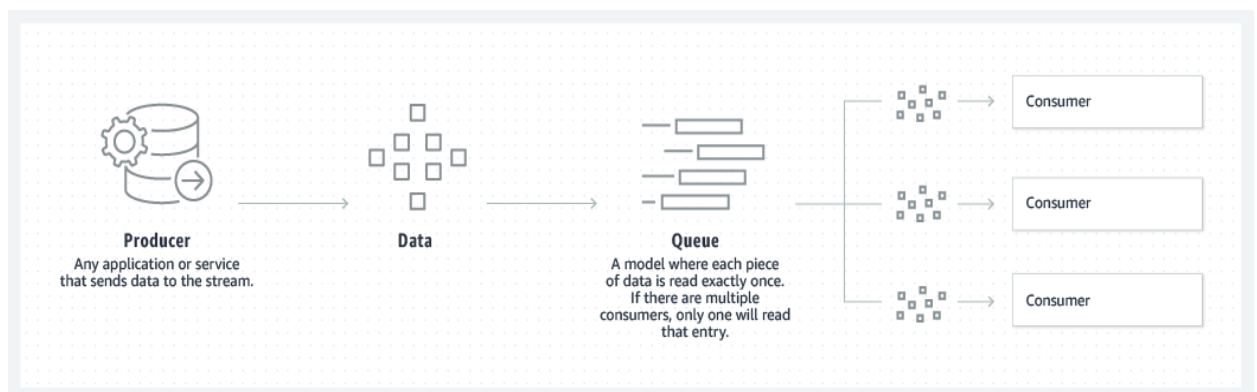
Kafka is used to build real-time streaming data pipelines and real-time streaming applications. A data pipeline reliably processes and moves data from one system to another, and a streaming application is an application that consumes streams of data. For example, if you want to create

a data pipeline that takes in user activity data to track how people use your website in real-time, Kafka would be used to ingest and store streaming data while serving reads for the applications powering the data pipeline. Kafka is also often used as a message broker solution, which is a platform that processes and mediates communication between two applications.

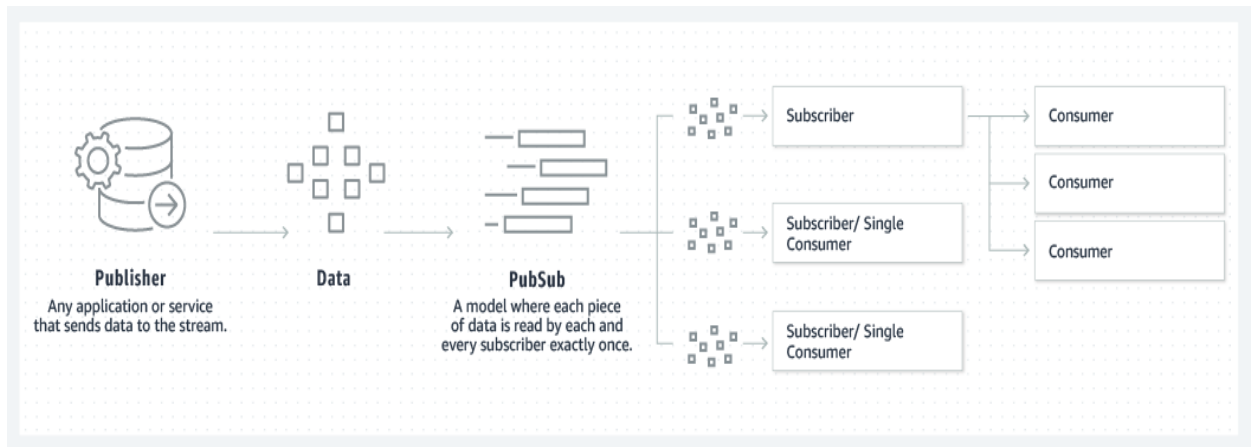
How does Kafka work?

Kafka combines two messaging models, queuing and publish-subscribe, to provide the key benefits of each to consumers. Queuing allows for data processing to be distributed across many consumer instances, making it highly scalable. However, traditional queues aren't multi-subscriber. The publish-subscribe approach is multi-subscriber, but because every message goes to every subscriber it cannot be used to distribute work across multiple worker processes. Kafka uses a partitioned log model to stitch together these two solutions. A log is an ordered sequence of records, and these logs are broken up into segments, or partitions, that correspond to different subscribers. This means that there can be multiple subscribers to the same topic and each is assigned a partition to allow for higher scalability. Finally, Kafka's model provides replayability, which allows multiple independent applications reading from data streams to work independently at their own rate.

Queuing



Publish-Subscribe



Benefits of Kafka's approach

- **Scalable**

Kafka's partitioned log model allows data to be distributed across multiple servers, making it scalable beyond what would fit on a single server.

- **Fast**

Kafka decouples data streams so there is very low latency, making it extremely fast.

- **Durable**

Partitions are distributed and replicated across many servers, and the data is all written to disk. This helps protect against server failure, making the data very fault-tolerant and durable.

Chapter-7

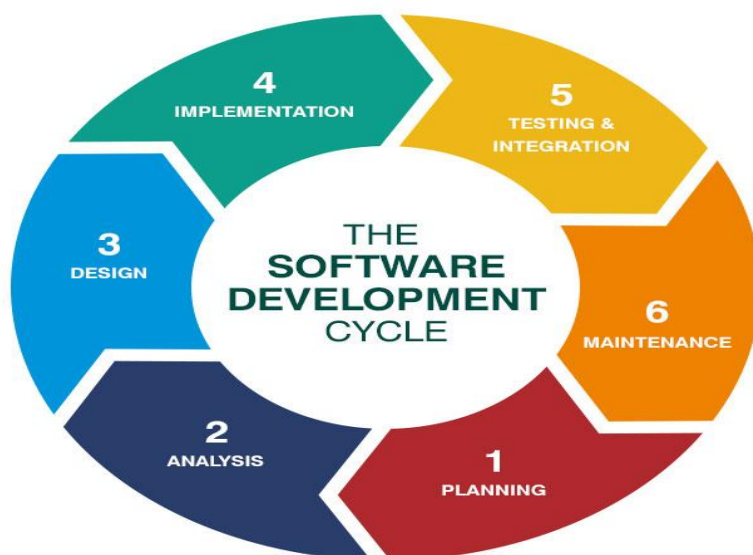
SDLC

7.1 Introduction to Software Development LifeCycle

Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality software's. The SDLC aims to produce a high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

- SDLC is the acronym of Software Development Life Cycle.
- It is also called as Software Development Process.
- SDLC is a framework defining tasks performed at each step in the software development process.
- ISO/IEC 12207 is an international standard for software life-cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software.

The following figure is a graphical representation of the various stages of a typical SDLC.



SDLC Models

There are various software development life cycle models defined and designed which are followed during the software development process. These models are also referred as Software

Development Process Models". Each process model follows a Series of steps unique to its type to ensure success in the process of software development.

Following are the most important and popular SDLC models followed in the industry –

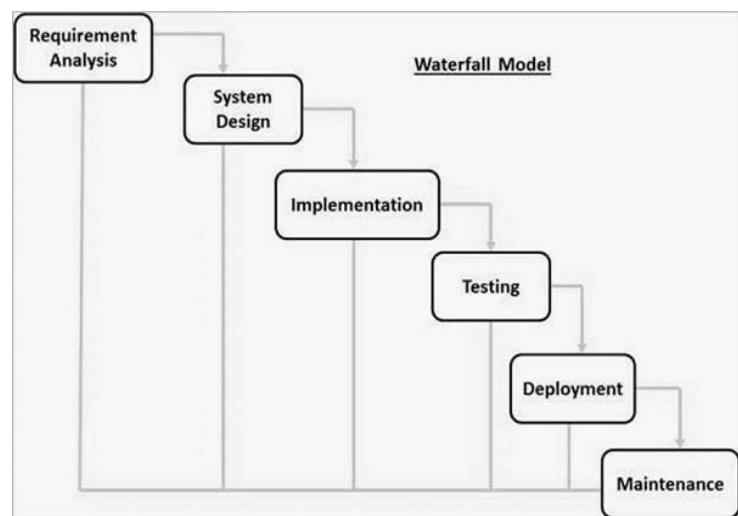
- Waterfall Model
- Iterative Model
- Spiral Model
- V-Model
- Big Bang Model

Other related methodologies are Agile Model, RAD Model, Rapid Application Development and Prototyping Models.

7.2 Waterfall Model

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.



The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

The sequential phases in Waterfall model are –

- **Requirement Gathering and analysis** – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

- **System Design** – The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** – With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** – All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** – There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

7.3 Agile Methodology

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like –

- Planning
- Requirements Analysis
- Design
- Coding

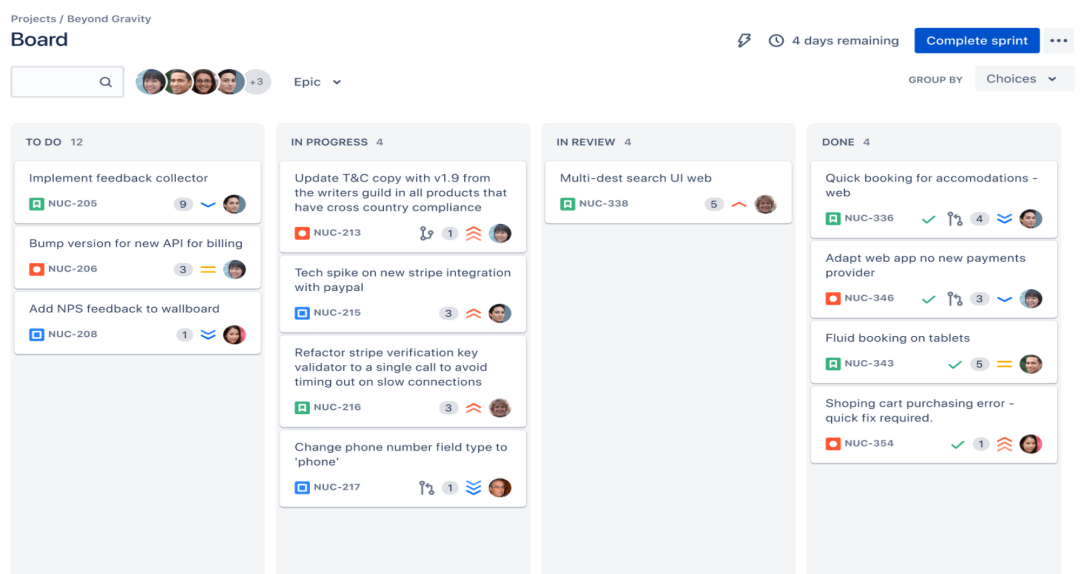


- Unit Testing and
- Acceptance Testing

7.4 Software Development Tools

- **Jira**

Jira Software is an agile project management tool that supports any agile methodology, be it scrum, Kanban, or your own unique flavor. From agile boards, backlogs, roadmaps, reports, to integrations and add-ons you can plan, track, and manage 40 all your agile software development projects from a single tool. Pick a framework to see how Jira Software can help your team release higher quality software, faster. Want to learn more about agile? Visit the Agile Coach, Atlassian's no-nonsense guide to agile development.



- **Postman**

Postman is a standalone software testing API (Application Programming Interface) platform to build, test, design, modify, and document APIs. It is a simple Graphic User Interface for sending and viewing HTTP requests and responses.



In this tool, nearly any functionality that any developer may need is embedded. This tool has the ability to make various types of HTTP requests like GET, POST, PUT, PATCH, and convert the API to code for languages like JavaScript and Python.

- **Docker**

Docker is a platform for packaging, deploying, and running applications. Docker applications run in containers that can be used on any system: a developer's laptop, systems on premises, or in the cloud.



Containerization is a technology that's been around for a long time, but it's seen new life with Docker. It packages applications as images that contain everything needed to run them: code, runtime environment, libraries, and configuration. Images run in containers, which are discrete processes that take up only as many resources as any other executable.

Containerization has enjoyed widespread adoption because of its

- Consistent test environment for development and QA.
- Cross-platform packages called images.
- Isolation and encapsulation of application dependencies.
- Ability to scale efficiently, easily, and in real time.

- **Gradle**

Gradle is an open source build automation tool that is based on the concept of Apache Maven and Apache Ant. It is capable of building almost any type of software. It is designed for the multi-project build, which can be quite large. It introduces a Java and Groovy-based DSL(Domain Specific Language) instead of XML (Extensible Markup Language) for declaring the project configuration. It uses a DAG (Directed Acyclic Graph) to define the order of executing the task.



Chapter-8

Project

8.1 Introduction

Background Verification System

Background Verification System is a web application which is used for the purpose of verifying the documents of the employees. The "Background Verification System" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and in some cases reduce the hardships faced by this existing system. Moreover this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

Every organization, whether big or small, has challenges to verify the employees for managing the integrity of the organization. The Background Verification System has 2 modules: Employer and Candidate. The Employer can View and Verify the documents and then Approve/Reject the application according to that. The Candidate have to upload the documents and then he can monitor the status of the background check.

8.2 Problem Statement

You are required to provide with a software solution regarding Background Verification System by which an Organization can easily Verify the documents of the recently hired candidates. You need to provide the solution which can help the employer to verify the documents of the candidates and then Approve/Reject the candidature of the candidate which the candidate can also monitor.

8.3 Objectives

The main objective of the Project on Background Verification System is to Verify the details and documents of the candidates.

It manages all the information about Candidates and helps the organization to verify it.

Functionalities provided by Faculty Management System are as follows:

- Provides the Upload Document functionality to candidates.
- Provides Approve/Reject Functionality to Employer.
- Candidate can monitor the status of his background check.

8.4 Project Architecture

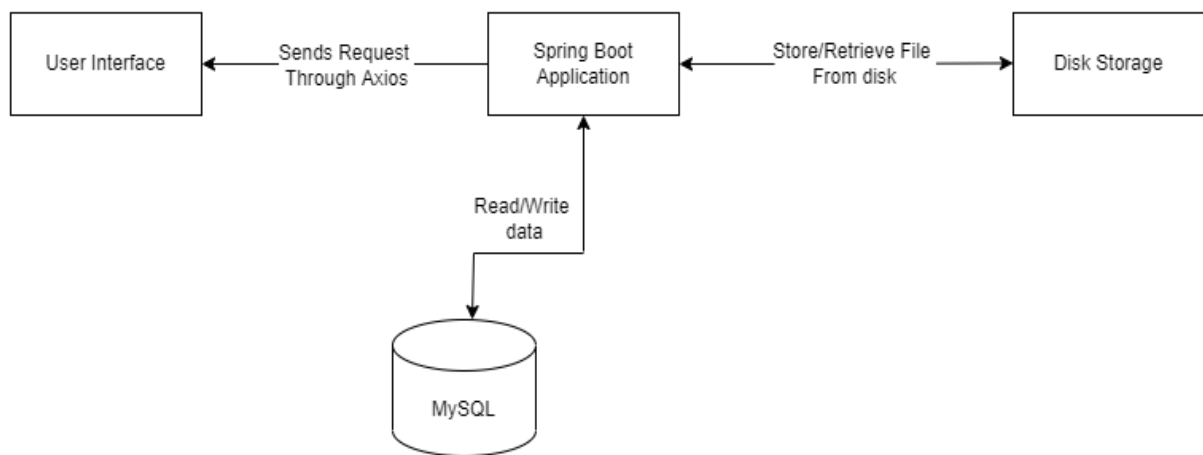


Figure 8.1

8.5 Tools and Technologies Used

The technology stack used in the project are as follows

- **Backend:** Spring Boot
- **Frontend:** ReactJS
- **Database:** MySQL
- **Testing:** Junit, Mockito
- **IDE:** IntelliJ
- **Build Tool:** Maven
- **Git Repository:** Github
- **Tools Used:** SonarQube, Jacoco, Postman

• 8.6 Use Case Diagram

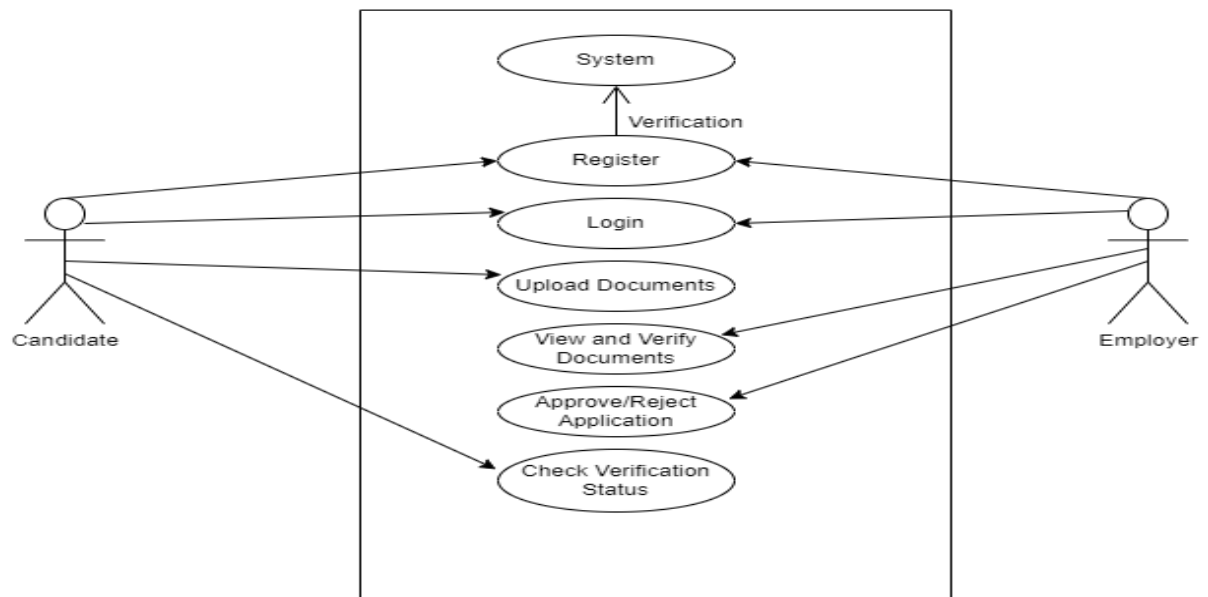


Figure 8.2

8.7 Activity Diagram

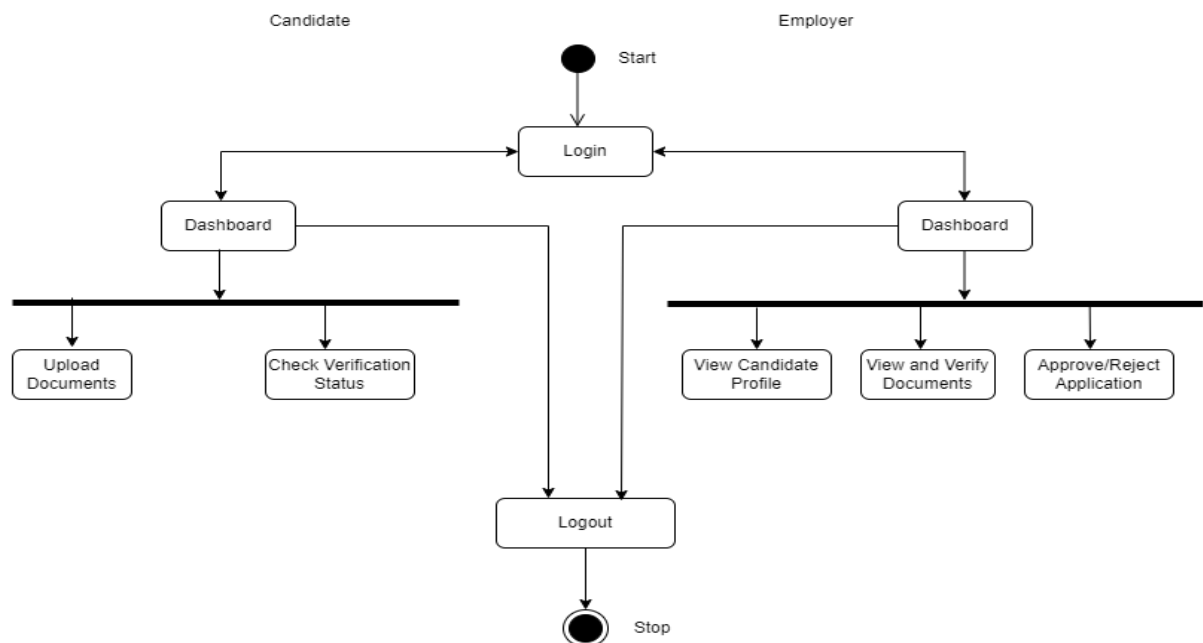


Figure 8.3

8.8 Sequence Diagram

- **Employer Sequence Diagram**

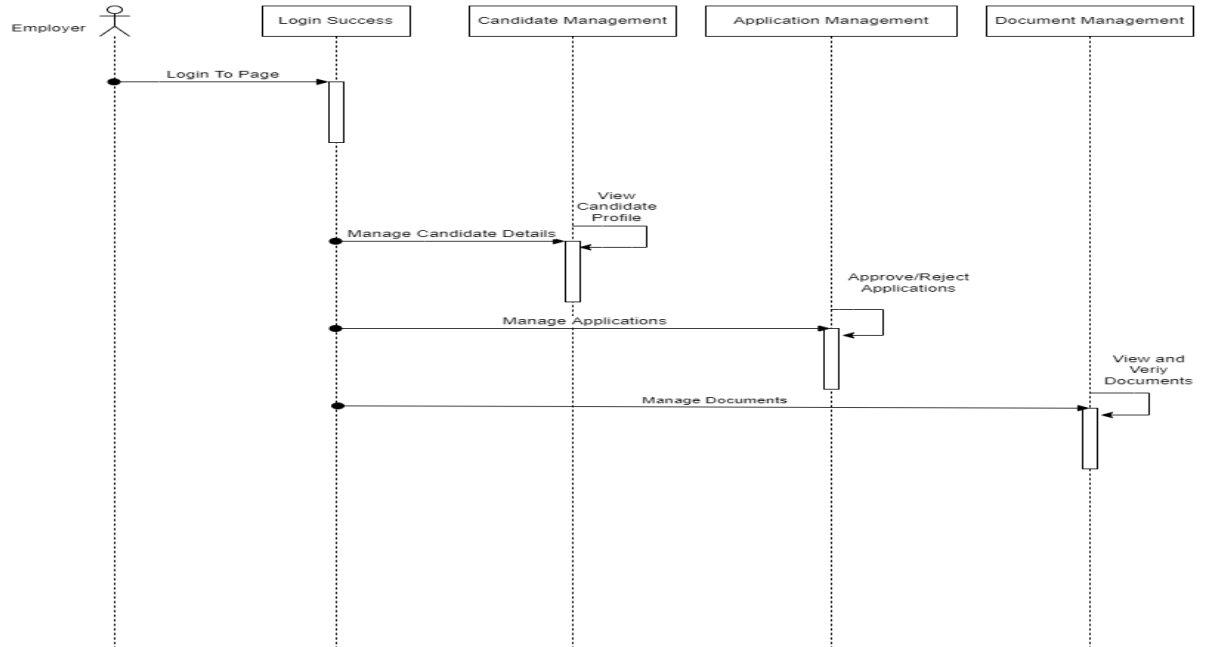


Figure 8.4

- **Candidate Sequence Diagram**

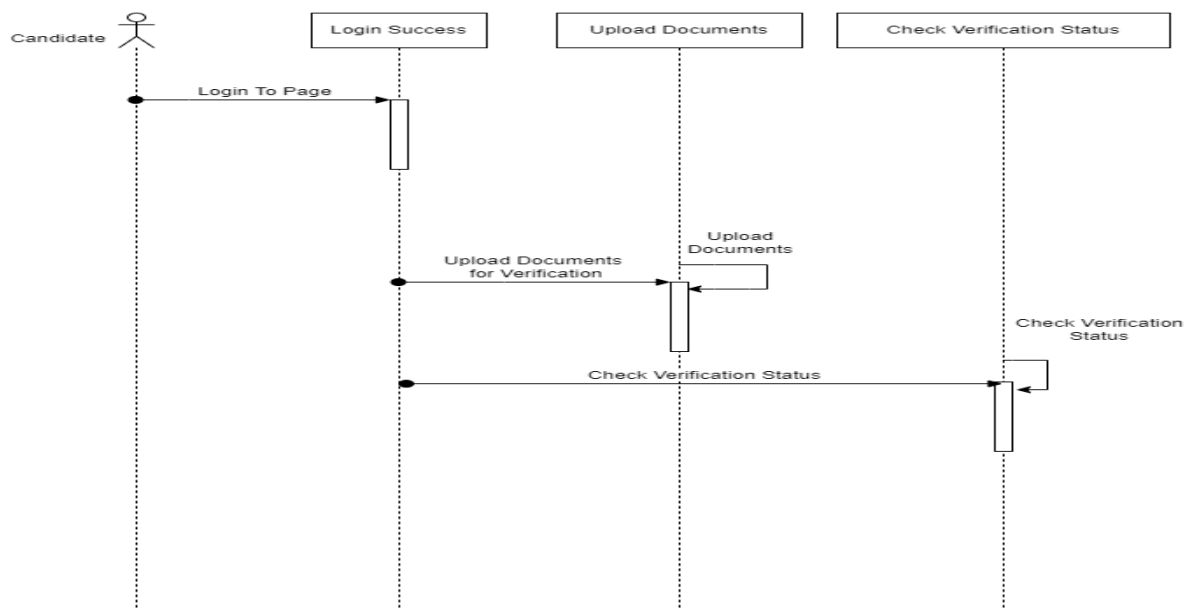


Figure 8.5

Chapter-9

Project Testing

9.1 Testing Methods Applied

Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance.

Unit Testing

This type of testing is performed by developers before the setup is handed over to the testing team to formally execute the test cases. Unit testing is performed by the respective developers on the individual units of source code assigned areas. The developers use test data that is different from the test data of the quality assurance team.

The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

Integration Testing

After each unit is thoroughly tested, it is integrated with other units to create modules or components that are designed to perform specific tasks or activities. These are then tested as group through integration testing to ensure whole segments of an application behave as expected (i.e, the interactions between units are seamless). These tests are often framed by user scenarios, such as logging into an application or opening files. Integrated tests can be conducted by either developers or independent testers and are usually comprised of a combination of automated functional and manual tests.

9.2 Test Cases

For not breaking the functionality of existing the system, we will test individual use cases and list down test cases for each of the use cases.

1. Login

➤ **Employer Login**

Test Case ID	Test	Expected Input	Expected Output	Actual Result	Status
1.	To Login into the Employer Dashboard	Username, Password	Login Successfully	Logged in Successfully (Employer Dashboard)	Pass
2.	Trying to Login with wrong Email/Password	Username, Password	Not able to Login	Not able to Login	Pass

➤ **Candidate Login**

Test Case ID	Test	Expected Input	Expected Output	Actual Result	Status
1.	To Login into the Candidate Dashboard	Username, Password	Login Successfully	Logged in Successfully (Candidate Dashboard)	Pass
2.	Trying to Login with wrong Email/Password	Username, Password	Not able to Login	Not able to Login	Pass

2. Register New Employer, Candidate

Test Case ID	Test	Expected Input	Expected Output	Actual Result	Status
1.	To Register new Employer and Candidate	Name, Email, , Department, Username and Password	Registered Successfully	Registered Successfully	Pass

2.	Trying to Register with any field being empty or already registered email	Name, Email, Department, Username and Password	Not able to Register	Not able to Register	Pass
----	---	--	----------------------	----------------------	------

3. Upload Document

Test Case ID	Test	Expected Input	Expected Output	Actual Result	Status
1.	To Upload Documents for Verification	File to upload	Uploaded Successfully	Uploaded Successfully	Pass
2.	Trying to Submit the form with any field being empty	File to upload	Not able to upload the file	Not able to upload the file	Pass

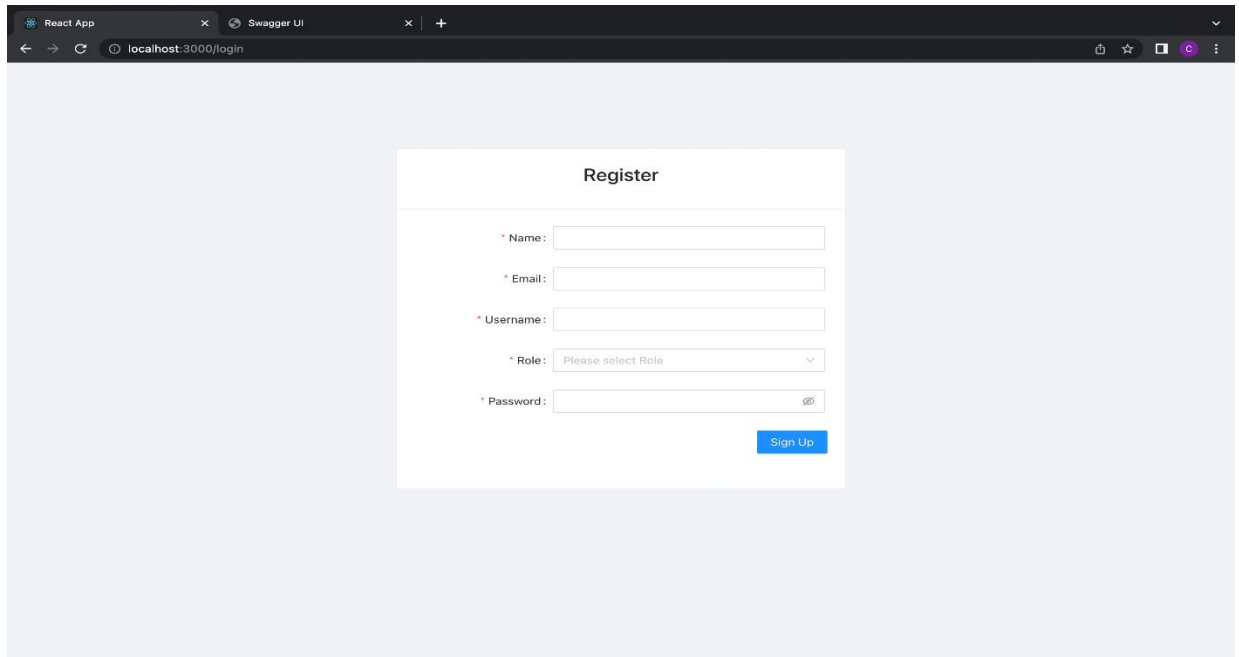
4. Approve/Reject Application

Test Case ID	Test	Expected Input	Expected Output	Actual Result	Status
1.	To Approve/Reject Application	Click the Approve/Reject Button	Approved/Rejected Successfully	Approved/Rejected Successfully	Pass
2.	Trying to Approve/Reject without clicking on the button	Click the Approve/Reject Button	Not able to Approve/Reject Application	Not able to Approve/Reject Application	Pass

Chapter-10

Snapshots of Project

1. Register Page

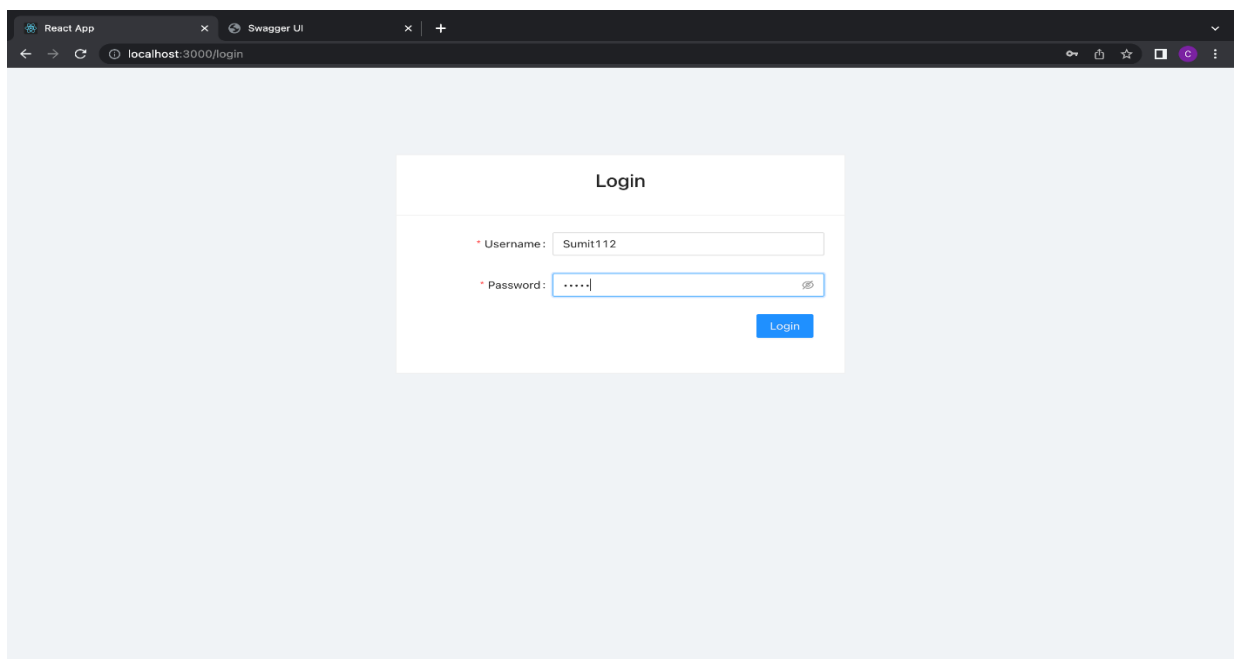


A screenshot of a web browser showing the 'Register' page. The browser has two tabs: 'React App' and 'Swagger UI'. The address bar shows 'localhost:3000/login'. The 'Register' form is centered on the page and contains the following fields:

- * Name:
- * Email:
- * Username:
- * Role:
- * Password:

A blue 'Sign Up' button is located at the bottom right of the form.

2. Login Page

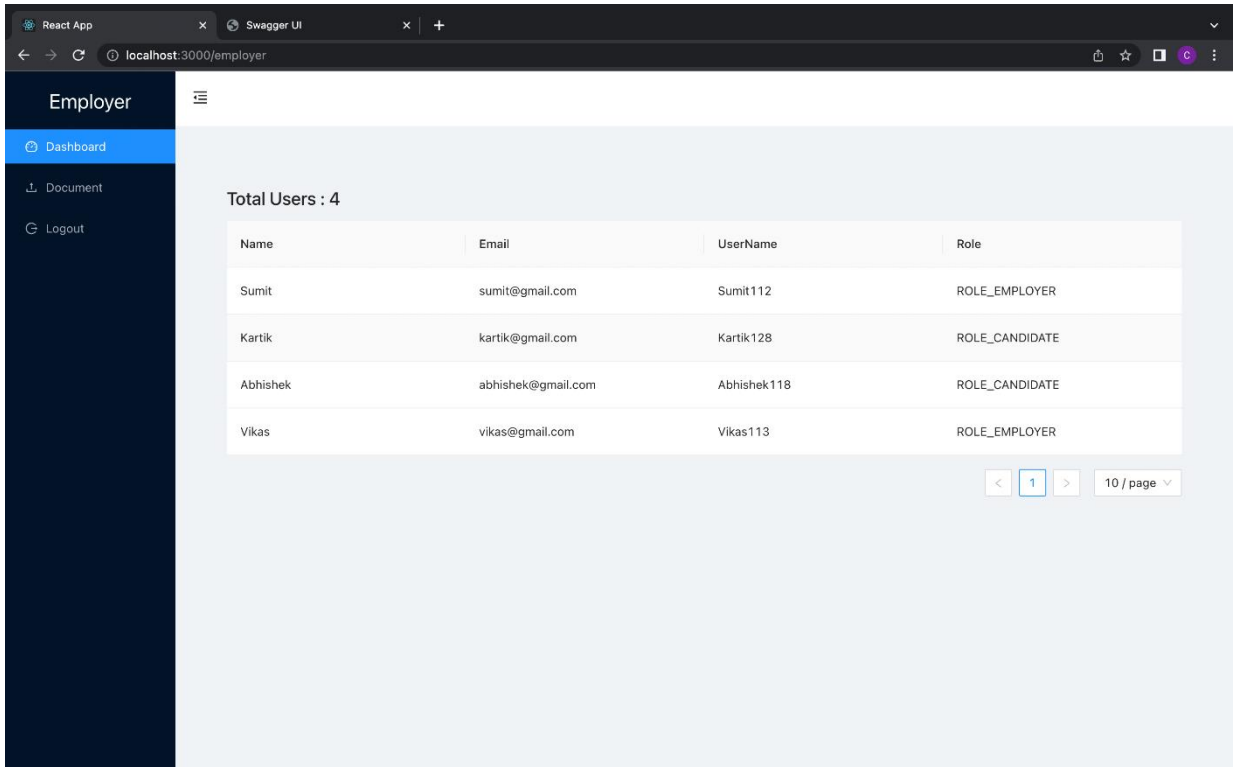


A screenshot of a web browser showing the 'Login' page. The browser has two tabs: 'React App' and 'Swagger UI'. The address bar shows 'localhost:3000/login'. The 'Login' form is centered on the page and contains the following fields:

- * Username:
- * Password:

A blue 'Login' button is located at the bottom right of the form.

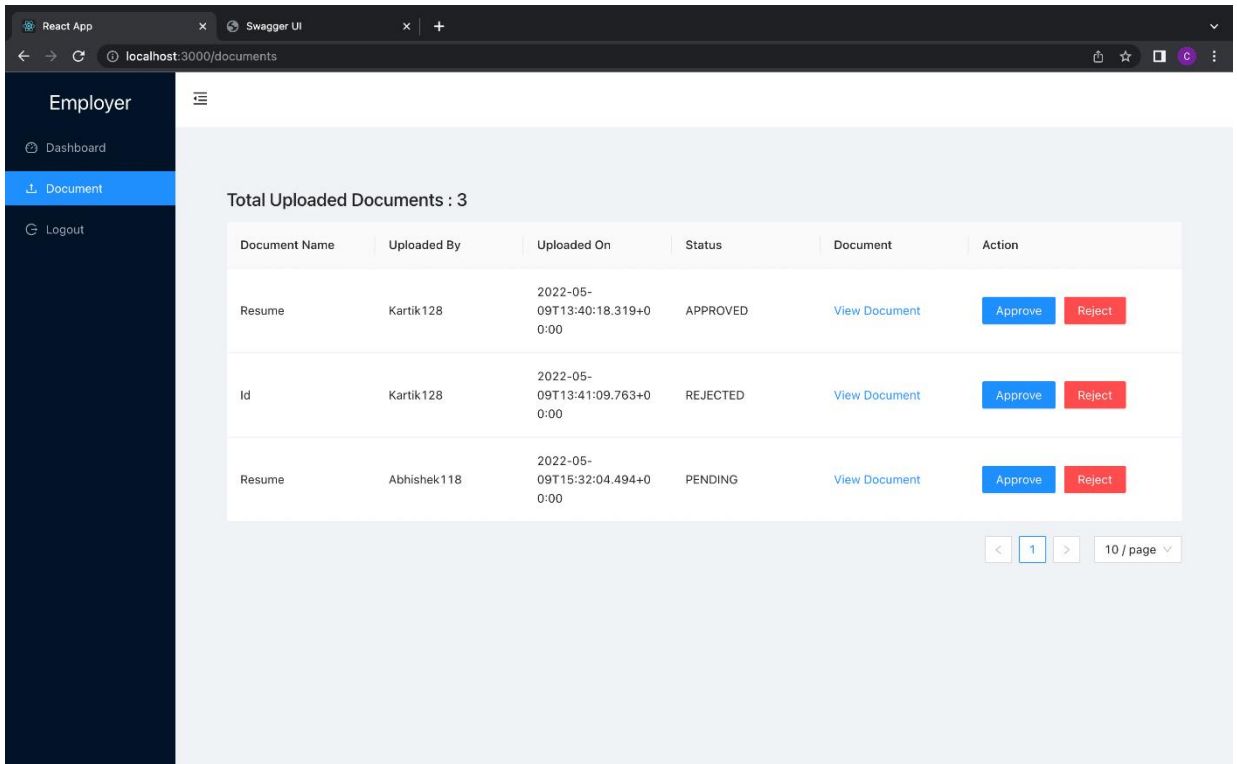
3. Employer Dashboard



The screenshot shows the Employer Dashboard in a web browser. The browser tabs include 'React App' and 'Swagger UI'. The address bar shows 'localhost:3000/employer'. The dashboard has a dark blue sidebar with 'Dashboard', 'Document', and 'Logout' options. The main content area displays 'Total Users : 4' and a table with user details. The table has columns for Name, Email, UserName, and Role. The data rows are: Sumit (sumit@gmail.com, Sumit112, ROLE_EMPLOYER), Kartik (kartik@gmail.com, Kartik128, ROLE_CANDIDATE), Abhishek (abhishek@gmail.com, Abhishek118, ROLE_CANDIDATE), and Vikas (vikas@gmail.com, Vikas113, ROLE_EMPLOYER). A pagination control at the bottom right shows '1' selected and '10 / page'.

Name	Email	UserName	Role
Sumit	sumit@gmail.com	Sumit112	ROLE_EMPLOYER
Kartik	kartik@gmail.com	Kartik128	ROLE_CANDIDATE
Abhishek	abhishek@gmail.com	Abhishek118	ROLE_CANDIDATE
Vikas	vikas@gmail.com	Vikas113	ROLE_EMPLOYER

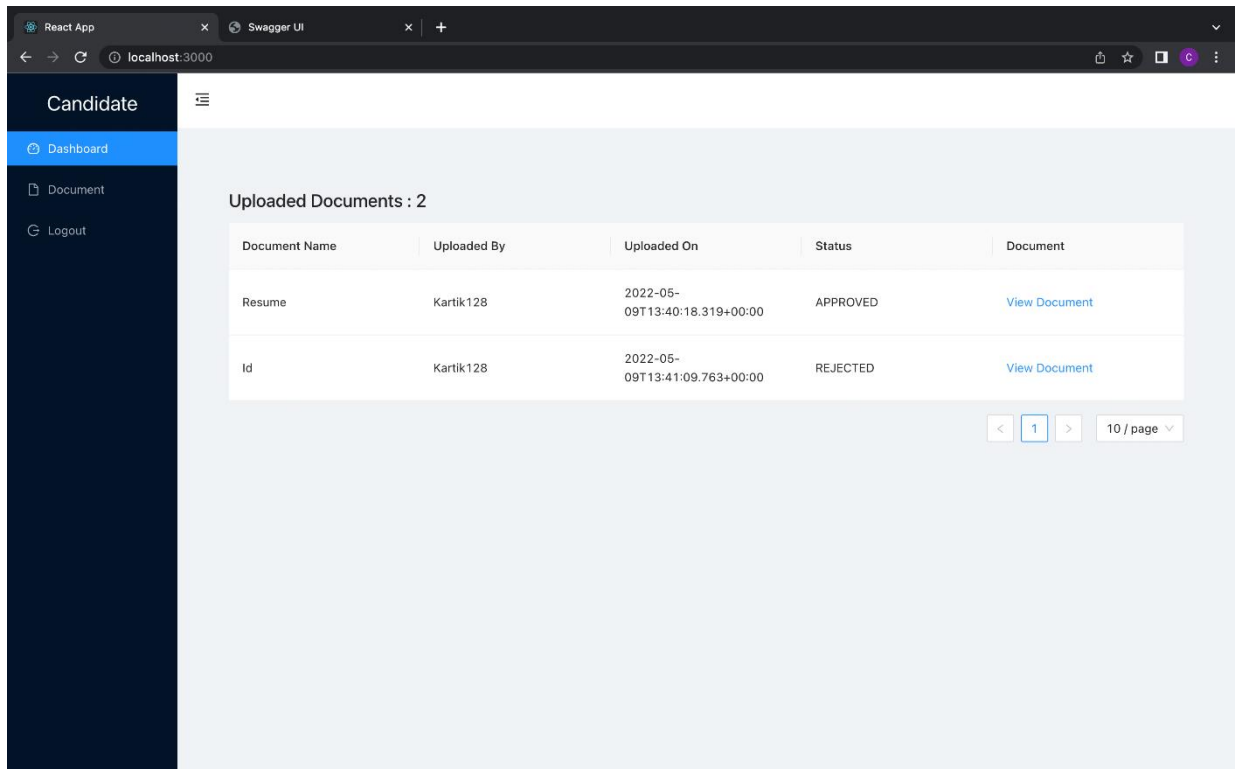
4. Employer Documents Page



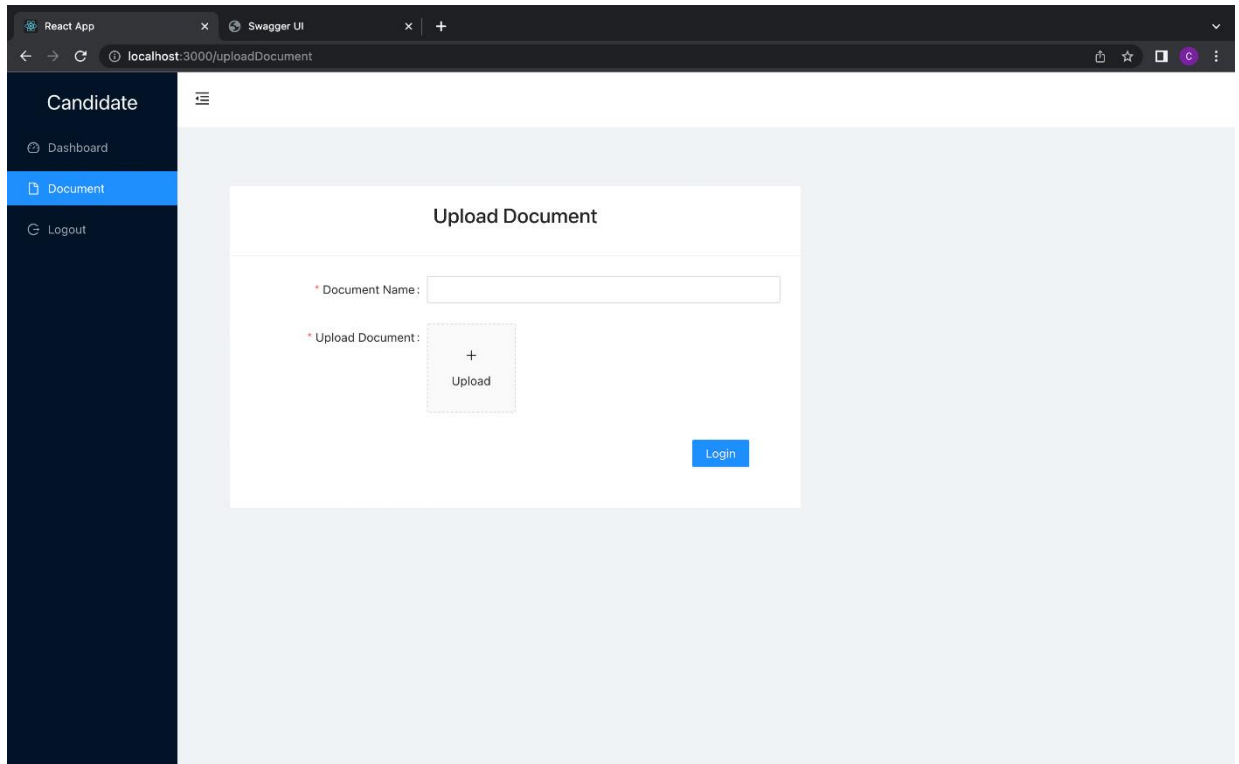
The screenshot shows the Employer Documents Page in a web browser. The browser tabs include 'React App' and 'Swagger UI'. The address bar shows 'localhost:3000/documents'. The dashboard has a dark blue sidebar with 'Dashboard', 'Document', and 'Logout' options. The main content area displays 'Total Uploaded Documents : 3' and a table with document details. The table has columns for Document Name, Uploaded By, Uploaded On, Status, Document, and Action. The data rows are: Resume (Kartik128, 2022-05-09T13:40:18.319+00:00, APPROVED, View Document, Approve, Reject), Id (Kartik128, 2022-05-09T13:41:09.763+00:00, REJECTED, View Document, Approve, Reject), and Resume (Abhishek118, 2022-05-09T15:32:04.494+00:00, PENDING, View Document, Approve, Reject). A pagination control at the bottom right shows '1' selected and '10 / page'.

Document Name	Uploaded By	Uploaded On	Status	Document	Action
Resume	Kartik128	2022-05-09T13:40:18.319+00:00	APPROVED	View Document	Approve Reject
Id	Kartik128	2022-05-09T13:41:09.763+00:00	REJECTED	View Document	Approve Reject
Resume	Abhishek118	2022-05-09T15:32:04.494+00:00	PENDING	View Document	Approve Reject

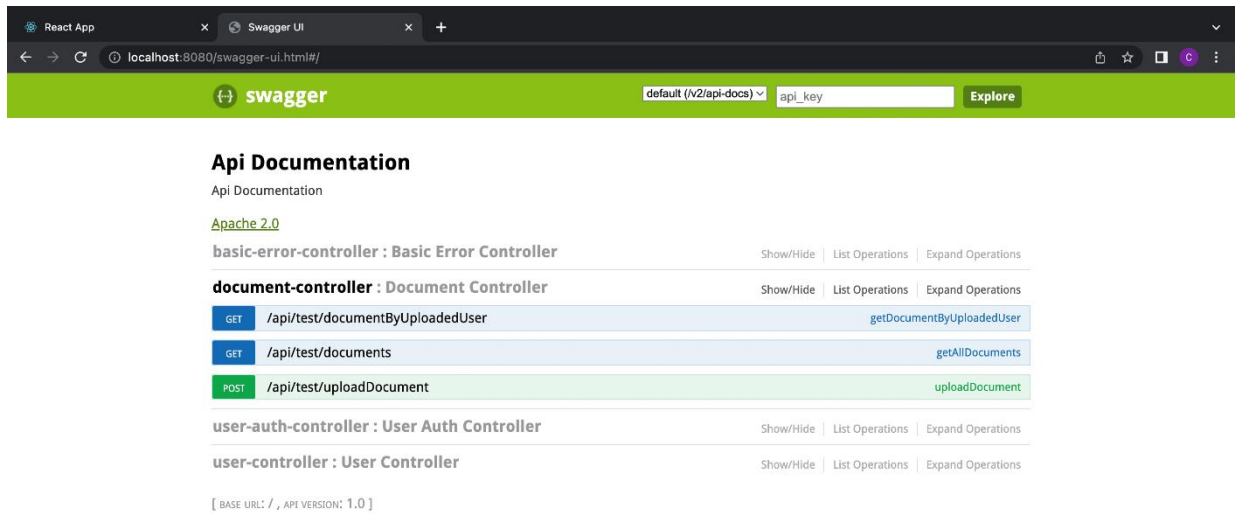
5. Candidate Dashboard



6. Candidate Upload Document Page



7. Swagger UI Page(API Documentation)



Chapter-11

Summary and Conclusion

Training and development are considered as a strategy for growth in the field of Computer Science and Information Technology. It is adopted by the organization to fill the gap between skills and future opportunities. These initial and advance training programs definitely enhance skills, improve efficiency, and productivity and growth opportunities for employees, mainly freshers. Skills, knowledge and attitudes are the basics for efficient running of a business through the human resources of an organization.

By improving the capabilities of employees, organizational capabilities can also be improved. In result, the structure of organization become flatter, in which there will be fewer levels between the top and the bottom employees. Many organizations provide coaching to their high-potential employees to develop their skills in creativity, thinking, innovation, vision, motivating others and brainstorming. Rather than putting the employees into management and expecting that they will develop their knowledge and skills on their own, organizations can systematically develop their skills through combination of these technical training and development programs. Evaluation of training must be appropriate for the persons and situations. The feedback from learners is important not only for instructors but also for confidence of the learner.

As a result of this internal/initial training, I am now aware of all these technologies mentioned, and also am ready to use that knowledge in building the upcoming project.

I was also given a series of evaluations which I passed with utmost accuracy to provide the proof of learning and the ability to use that knowledge practically.

Chapter-12

Bibliography

- "What is version control | Atlassian Git Tutorial", Atlassian. [Online]. Available: <https://www.atlassian.com/git/tutorials/what-is-version-control>. [Accessed: 10- Mar- 2022].
- "Online Tutorials Library" Tutorialspoint.com. [Online]. Available: www.tutorialspoint.com [Accessed: 15- Mar- 2022].
- "Tutorials - Javatpoint", www.javatpoint.com, 2022. [Online]. Available: <https://www.javatpoint.com/>. [Accessed: 15- Mar- 2022].
- "W3Schools Free Online Web Tutorials", w3schools.com, 2022. [Online]. Available: <https://www.w3schools.com/>. [Accessed: 22- Mar- 2022].
- "Learn Spring Tutorial – javatpoint", www.javatpoint.com, 2022. [Online]. Available: <https://www.javatpoint.com/spring-tutorial>. [Accessed: 22- Mar- 2022].
- "MySQL Tutorial - Learn MySQL Fast, Easy and Fun.", MySQL Tutorial, 2022. [Online]. Available: <https://www.mysqltutorial.org/>. [Accessed: 22- Mar- 2022].
- "MongoDB Tutorial for Beginners", Guru99, 2022. [Online]. Available: <https://www.guru99.com/mongodb-tutorials.html>. [Accessed: 22- Mar- 2022].
- "Tutorial: Intro to React – React", Reactjs.org, 2022. [Online]. Available: <https://reactjs.org/>. [Accessed: 01- Apr- 2022].
- "Redux Essentials, Part 1: Redux Overview and Concepts | Redux", Redux.js.org, 2022. [Online]. Available: <https://redux.js.org/tutorials/essentials/part-1-overview-concepts>. [Accessed: 01- Apr- 2022].
- "Apache Kafka - Apache Kafka". [Online]. Available: <https://kafka.apache.org/>. [Accessed: 05- Apr- 2022].
- "Software Engineering | Classical Waterfall Model – GeeksforGeeks", GeeksforGeeks, 2022. [Online]. Available: <https://www.geeksforgeeks.org/software-engineering-classical-waterfall-model/>. [Accessed: 05- Apr- 2022].
- "Flowchart Maker & Online Diagram Software", app.diagrams.net, 2022. [Online]. Available: <https://app.diagrams.net/>. [Accessed: 10- Apr- 2022]
- "Home – Docker", Docker 2022. [Online]. Available: <https://www.docker.com/>. [Accessed: 10- Apr- 2022].