

SMART RECRUITER WEBSITE

Submitted in partial fulfillment of the requirements

For the award of the degree of

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

Submitted by

R.S. SATHISH (21SUCS18)

Under the Guidance of

Dr.A.SHARMISTA MCA.,B.ED.,M.Phil.,Ph.D.,



DEPARTMENT OF COMPUTER SCIENCE

THIAGARAJAR COLLEGE (AUTONOMOUS)

(Affiliated to Madurai Kamaraj University)

Re-Accredited with “A++ Grade” by NAAC

Ranked 18th in NIRF

Madurai-625009

APRIL-2024

SMART RECRUITER WEBSITE

Submitted in partial fulfillment of the requirements

For the award of the degree of

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

Submitted by

R.S. SATHISH (21SUCS18)

Under the Guidance of

Dr.A.SHARMISTA MCA.,B.ED.,M.Phil.,Ph.D.,



**DEPARTMENT OF COMPUTER SCIENCE
THIAGARAJAR COLLEGE (AUTONOMOUS)**

(Affiliated to Madurai Kamaraj University)

Re-Accredited with “A++ Grade” by NAAC

Ranked 18th in NIRF

Madurai-625009

APRIL-2024



THIAGARAJAR COLLEGE (AUTONOMOUS)

(Affiliated to Madurai Kamaraj University)

Re-Accredited with “A++ Grade” by NAAC

Ranked 18th in NIRF

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that the project work entitled “**SMART RECRUITER WEBSITE**” is the project work, done by **R.S. SATHISH (21SUCS18)** in the partial fulfillment of the requirement for the award of Bachelor of Science in Computer Science during final viva voce examination held on _____.

Project Guide
(Dr. A. SHARMISTA)

Head of the Department
(Dr. G. RAKESH)

External Examiner

R.S. SATHISH (21SUCS18)

III B.Sc (Computer Science),

Thiagarajar College (Autonomous),

Madurai- 625009.

DECLARATION

I hereby declare that the project work entitled “**SMART RECRUITER WEBSITE**” is submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Science in Computer Science. This is the record of the original work done by us during the period of project work and has not been presented anywhere for any award or title.

Place: Madurai

R.S.SATHISH

Date:

(21SUCS18)

ACKNOWLEDGEMENTS

I extend my sincere thanks to the almighty for giving me physical, and mental strength and ability for the effective completion of the project.

I am grateful to thank our principal **Dr. D. PANDIARAJA M.Sc., M.Phil., P.G.D.CA., B.Ed., Ph.D.** Thiagarajar College, Madurai for his encouragement to take up this project.

I like to express my profound gratitude for the support offered by the Head of the Department of Computer Science **Dr. G. RAKESH M.C.A., M.Phil., Ph.D.**, I am also thankful to her who guided and encouraged me in this project work.

I extend my sincere thanks to my project guide **Dr. A. SHARMISTA M.C.A., B.Ed., M.Phil., Ph.D.**, I will be grateful to her for having faith in my ability and entrusting me with this project.

I thank all the faculty members of the Computer Science department for their cooperation. I also personally thank our family members and friends for their continuous encouragement.

CONTENTS

CONTENTS

S.NO	TITLES	PAGE NO
1.	INTRODUCTION	1
	1.1 Abstract	2
	1.2 Aims and Objectives	3
2.	SYSTEM ANALYSIS	4
	2.1 Existing System	5
	2.2 Proposed System	6
	2.3 Feasibility Study	7
3.	SYSTEM SPECIFICATION	10
	3.1 Hard Specification	11
	3.2 Software Specification	11
	3.3 Software Description	12
4.	PROJECT DESCRIPTION	19
5.	SYSTEM DESIGN	24
	4.1 Input Design	26
	4.2 Output Design	27
	4.3 Database Design	28
	4.4 Data Flow Diagram	31
6.	SYSTEM TESTING & IMPLEMENTATION	37
	5.1 System Testing	38
	5.2 Implementation	43
7.	SAMPLES	46
	7.1 Coding	47
	7.2 Screenshots	61
8.	CONCLUSION	68
9.	FUTURE ENHANCEMENT	70
10.	BIBLIOGRAPHY	72
	10.1 Book References	73
	10.2 Web References	73

INTRODUCTION

1. INTRODUCTION

1.1 ABSTRACT:

“Smart Recruiter Website” is a website that acts as an intermediate for Job Recruitment between the organization and the job applicants.

The purpose of this website is to organize the recruitment process for the organization and also manage the job applicants who applied for the specified job roles.

This website offers many features that make it user-friendly such as a filtering option for job applicants based on their skills, experience, educational qualifications, etc..

It is an applicant-tracking management system, that helps an organization in its hiring process with features like job creation, managing applicants (applied, recommended, hired, declined, withdrawn), setting up an online interview, sending rejection or accepting emails to hundreds of candidates in a single click, managing the record of applicants with great data visualization, job details, and application forms for end-users.

Develop a user-friendly interface for recruiters to manage job postings, candidate profiles, and communication efficiently.

Implement AI-powered algorithms for resume screening and candidate matching based on job requirements and qualifications.

Integrate communication tools such as chatbots or automated emails to keep candidates informed about their application status.

Incorporate data analytics to track recruitment metrics such as time-to-hire, cost-per-hire, and candidate quality.

Provide training and support for recruiters to adapt to the new system and maximize its effectiveness.

1.2 AIMS AND OBJECTIVES:

The main aim and objectives are

- To streamline the recruitment process through automation and intelligent algorithms.
- To improve the efficiency of candidate sourcing, screening, and selection.
- To enhance the candidate experience by providing personalized and timely communication.
- To reduce recruitment costs and time-to-fill positions.
- To ensure fair and unbiased candidate evaluation through data-driven decision-making.
- To leverage technology to identify and attract top talent in the industry.
- To optimize the allocation of human resources in the recruitment process by automating repetitive tasks.
- To create a scalable and adaptable recruitment solution that can cater to the evolving needs of the organization.
- To enhance the employer brand by showcasing a modern and innovative approach to recruitment.

SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM:

- Manual processes, such as sorting through resumes, scheduling interviews, and communicating with candidates, can be time-consuming.
- Traditional methods of job recruitment, such as posting ads in newspapers or relying on personal networks, have a limited reach, which can result in a smaller pool of candidates, potentially missing out on highly qualified individuals.
- Manual processes often lead to higher recruitment costs, which include expenses related to printing and distributing job advertisements, conducting physical interviews, and managing administrative tasks.
- Manual data entry and record-keeping increase the risk of human errors which can lead to inaccuracies in candidate information.

DISADVANTAGES:

- Delays in the recruitment process may lead to the loss of qualified candidates to competitors.
- Inaccurate data can lead to misunderstandings, miscommunication, and errors in decision-making.
- Lack of streamlined communication and feedback during the recruitment process.
- Manual processes can become increasingly challenging to manage as the volume of job applications grows.
- Traditional recruitment methods may incur higher costs, including expenses related to advertising in print media, travel for in-person interviews, and administrative overhead.

2.2 PROPOSED SYSTEM:

- Automated recruitment processes reduce manual effort, speeding up the recruitment cycle and allowing recruiters to focus on strategic aspects of hiring.
- Utilizing Machine Learning (ML) algorithms to analyze resumes objectively. This not only speeds up the screening process but also minimizes bias by focusing on skills and qualifications.
- Integration of scheduling for online interviews and coordination of interview timings.
- Implementation of a platform for conducting remote video interviews.
- Implementing a feedback loop and analytics for continuous improvement.

ADVANTAGES:

- Improved organization, faster resume screening, and simplified collaboration among hiring team members.
- Efficient and unbiased shortlisting of candidates based on specific criteria, saving time and reducing human bias.
- Reduces communication delays, streamlines the interview process, and enhances the candidate experience.
- Enables organizations to learn from past recruitment processes, refine strategies, and adapt to changing market conditions.
- Allows for asynchronous interviews, providing flexibility for both candidates and interviewers.

2.3 FEASIBILITY STUDY

The objective of the feasibility study is not only to solve the problem but also to acquire a sense of its scope. The reason for doing this is to identify the most beneficial project to the organization.

Three key considerations involved in the feasibility analysis are

1. Technical Feasibility
2. Financial Feasibility
3. Operating Feasibility

Technical Feasibility

Technical feasibility is the study of the software and how it is included in the study of our project.

Regarding this, there are some technical issues that should be noted they are as follows:

- Is the necessary technique available and how it is suggested and acquired?
- Does the proposed equipment have the technical capacity to hold the data required using the new system?
- Can this system be expanded after this project development?
- Is there a technique guarantees of accuracy, reliability in case of access of data?

The technical issues are raised during the feasibility study of investigating our system. Thus, the technical consideration evaluates the hardware requirements, software etc. This system uses PHP as front end and MySQL as back end. They also provide sufficient memory to hold and process the data. As the organization is going to install all the process in the system it is cheap and efficient technique. This system technique accepts the entire request made by the user and the response is done without failure and delay. It is a study about the resources available and how they are achieved as an acceptable system. It is an essential process for analysis and definition of conducting a parallel assessment of technical feasibility.

Economical Feasibility

An organization makes good investment on the system. So, they should be worthful for the amount they spend in the system.

Always the financial benefit are equals or less the cost of the system, but should not exceed the cost:

- The cost of investment is analyzed for the entire system.
- The cost of Hardware and Software is also noted.
- Analyzing the way in which the cost can be reduced.

Every organization want to reduce their cost but at the same time quality of the service should also be maintained. This system is developed according the estimation of the cost made by the concern. In this project, the proposed system will definitely reduce the cost and also the manual work is reduced and speed of work is also increased.

Operational Feasibility

Proposed project will be beneficial only when they are turned into an information system and to meet the organization operating requirements.

The following issues are considered for the operation:

- Does this system provide sufficient support for the user and the management?
- What is the method that should be used in this project?
- Have the users been involved in the planning and development of the projects?
- Will the proposed system cause any harm, bad result, loss of control and accessibility of the system will lost?

Issues that may be a minor problem will sometimes cause major problem in the operation. It is the measure of how people can able to work with the system. Finding out the minor issues that may be the initial problem of the system. It should be a user-friendly environment. All these aspects should be kept in mind and steps should be taken for developing the project carefully.

Regarding the project, the system is very much supported and friendly for the user. The methods are defined in an effective manner and proper conditions are given in order to avoid the harm or loss of data. It is designed in GUI interface, as working will be easier and flexible for the user. These are three basic feasibility studies that are done in every project.

SYSTEM SPECIFICATION

3. SYSTEM SPECIFICATION

3.1 HARDWARE SPECIFICATION:

- Processor : 12th Gen Intel(R) Core(TM) i5-1235U
- Processor Speed : 1.30 GHz
- Hard disk : 378GB
- Ram : 16GB

3.2 SOFTWARE SPECIFICATION:

- Operating System : Windows 11
- Frontend : React JS
- Backend : Node JS, Express JS, MongoDB
- Browser : Google Chrome Version 97.0

3.3 SOFTWARE DESCRIPTION:

3.3.1 Intorduction to React JS:

React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces based on components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies.

React can be used to develop single-page, mobile, or server-rendered applications with frameworks like Next.js. Because React is only concerned with the user interface and rendering components to the DOM, React applications often rely on libraries for routing and other client-side functionality. A key advantage of React is that it only rerenders those parts of the page that have changed, avoiding unnecessary rerendering of unchanged DOM elements

Declarative

React adheres to the declarative programming paradigm. Developers design views for each state of an application, and React updates and renders components when data changes. This is in contrast with imperative programming.

Components

- React code is made of entities called components. These components are modular and reusable.
- React applications typically consist of many layers of components. The components are rendered to a root element in the DOM using the React DOM library.
- When rendering a component, values are passed between components through props (short for "properties"). Values internal to a component are called its state.
- The two primary ways of declaring components in React are through function components.

Function components

Function components are declared with a function (using JavaScript function syntax or an arrow function expression) that accepts a single "props" argument and returns JSX. From React v16.8 onwards, function components can use state with the `useState` Hook.

React Hooks

On February 16, 2019, React 16.8 was released to the public, introducing React Hooks. Hooks are functions that let developers "hook into" React state and lifecycle features from function components. Notably, Hooks do not work inside classes — they let developers use more features of React without classes.

React provides several built-in Hooks such as `useState`, `useContext`, `useReducer`, `useMemo` and `useEffect`. Others are documented in the Hooks API Reference. `useState` and `useEffect`, which are the most commonly used, are for controlling state and side effects respectively.

Rules of hooks

There are two rules of Hooks which describe the characteristic code patterns that Hooks rely on:

- "Only Call Hooks at the Top Level" — Don't call hooks from inside loops, conditions, or nested statements so that the hooks are called in the same order each render.
- "Only Call Hooks from React Functions" — Don't call hooks from plain JavaScript functions so that stateful logic stays with the component.

Although these rules can't be enforced at runtime, code analysis tools such as linters can be configured to detect many mistakes during development.

Lifecycle methods

Lifecycle methods for class-based components use a form of hooking that allows the execution of code at set points during a component's lifetime.

- **ShouldComponentUpdate** allows the developer to prevent unnecessary re-rendering of a component by returning false if a render is not required.
- **componentDidMount** is called once the component has "mounted" (the component has been created in the user interface, often by associating it with a DOM node). This is commonly used to trigger data loading from a remote source via an API.
- **componentWillUnmount** is called immediately before the component is torn down or "unmounted". This is commonly used to clear resource-demanding dependencies to the component that will not simply be removed with the unmounting of the component (e.g., removing any `setInterval()` instances that are related to the component, or an "eventListener" set on the "document" because of the presence of the component)
- **render** is the most important lifecycle method and the only required one in any component. It is usually called every time the component's state is updated, which should be reflected in the user interface.

Virtual DOM

Another notable feature is the use of a virtual Document Object Model or Virtual DOM. React creates an in-memory data-structure cache, computes the resulting differences, and then updates the browser's displayed DOM efficiently. This process is called reconciliation. This allows the programmer to write code as if the entire page is rendered on each change, while React only renders the components that actually change. This selective rendering provides a major performance boost.

3.3.2 Introduction to MongoDB:

MongoDB is a source-available, cross-platform, document-oriented database program. Classified as a NoSQL database product, MongoDB utilizes JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and current versions are licensed under the Server Side Public License (SSPL). MongoDB is a member of the MACH Alliance.

History:

- The American software company 10gen began developing MongoDB in 2007 as a component of a planned platform-as-a-service product. In 2009, the company shifted to an open-source development model and began offering commercial support and other services. In 2013, 10gen changed its name to MongoDB Inc.
- On October 20, 2017, MongoDB became a publicly traded company, listed on NASDAQ as MDB with an IPO price of \$24 per share.
- On November 8, 2018 with the stable release 4.0.4, the software's license changed from AGPL 3.0 to SSPL.
- On October 30, 2019, MongoDB teamed with Alibaba Cloud to offer Alibaba Cloud customers a MongoDB-as-a-service solution. Customers can use the managed offering from Alibaba's global data centers.

Main features

Ad-hoc queries:

MongoDB supports field, range query and regular-expression searches. Queries can return specific fields of documents and also include user-defined JavaScript functions.

Indexing:

Fields in a MongoDB document can be indexed with primary and secondary indices.

Replication:

MongoDB provides high availability with replica sets. A replica set consists of two or more copies of the data. Each replica-set member may act in the role of primary or secondary replica at any time.

All writes and reads are done on the primary replica by default. Secondary replicas maintain a copy of the data of the primary using built-in replication. When a primary replica fails, the replica set automatically conducts an election process to determine which secondary should become the primary.

Secondaries can optionally serve read operations, but that data is only eventually consistent by default.

If the replicated MongoDB deployment only has a single secondary member, a separate daemon called an arbiter must be added to the set. It has the single responsibility of resolving the election of the new primary.

As a consequence, an ideal distributed MongoDB deployment requires at least three separate servers, even in the case of just one primary and one secondary.

Load balancing:

MongoDB scales horizontally using sharding. The user chooses a shard key, which determines how the data in a collection will be distributed. The data is split into ranges (based on the shard key) and distributed across multiple shards, which are masters with one or more replicas. Alternatively,

the shard key can be hashed to map to a shard—enabling an even data distribution.

MongoDB can run over multiple servers, balancing the load or duplicating data to keep the system functional in case of hardware failure.

File storage:

MongoDB can be used as a file system, called GridFS, with load-balancing and data-replication features over multiple machines for storing files.

This function, called a grid file system, is included with MongoDB drivers. MongoDB exposes functions for file manipulation and content to developers.

GridFS can be accessed using the mongofiles utility or plugins for Nginx and lighttpd. GridFS divides a file into parts, or chunks, and stores each of those chunks as a separate document.

Aggregation:

MongoDB provides three ways to perform aggregation: the aggregation pipeline, the map-reduce function and single-purpose aggregation methods.

Map-reduce can be used for batch processing of data and aggregation operations. However, according to MongoDB's documentation, the aggregation pipeline provides better performance for most aggregation operations.

The aggregation framework enables users to obtain results similar to those returned by queries that include the SQL GROUP BY clause. Aggregation operators can be strung together to form a pipeline, analogous to Unix pipes.

The aggregation framework includes the \$lookup operator, which can join documents from multiple collections, as well as statistical operators such as standard deviation.

Server-side JavaScript execution:

JavaScript can be used in queries, aggregation functions (such as MapReduce) and sent directly to the database to be executed.

Capped collections:

MongoDB supports fixed-size collections called capped collections. This type of collection maintains insertion order and, once the specified size has been reached, behaves like a circular queue.

Transactions:

MongoDB supports multi-document ACID transactions since the 4.0 release in June 2018.

MongoDB Community Server

As of October 2018, MongoDB is released under the Server Side Public License (SSPL), a non-free license developed by the project. It replaces the GNU Affero General Public License, and is nearly identical to the GNU General Public License version 3, but requires that those making the software publicly available as part of a "service" must make the service's entire source code (insofar that a user would be able to recreate the service themselves) available under this license. By contrast, the AGPL only requires the source code of the licensed software to be provided to users when the software is conveyed over a network. The SSPL was submitted for certification to the Open Source Initiative but later withdrawn. In January 2021, the Open Source Initiative stated that SSPL is not an open source license. The language drivers are available under an Apache License. In addition, MongoDB Inc. offers proprietary licenses for MongoDB. The last versions licensed as AGPL version 3 are 4.0.3 (stable) and 4.1.4.

PROJECT DESCRIPTION

4. PROJECT DESCRIPTION

4.1 MODULES:

Organization Module :

- Register
- Login
- Home
- Employees
- Jobs
- Candidates
- Statistics
- Settings

Applicants Module :

- Register
- Login
- Home
- View Jobs
- Apply Jobs
- Receive Mail
- Settings

4.2 MODULE DESCRIPTION:

1. ORGANIZATION MODULE :

a. Register:

Upon signing up for the first time, users will receive a verification email to confirm their registration. Clicking the verification link within the email will securely redirect them to the login page, allowing seamless access to their account. This process ensures the security and validity of user accounts while facilitating a smooth login experience.

b. Login:

The login page serves as the gateway to secure access, seamlessly verifying the authenticity of the entered email address and password. By conducting a thorough validation process, the system ensures that only valid email credentials gain entry. Once verified, users are granted access to the home page.

c. Home:

On the home page, the organizations can seamlessly register and upon registration, organizations are seamlessly directed to their dedicated details page, facilitating the collection of essential information. Once registered, explore the efficiency as all associated departments within the organization are dynamically showcased, providing a comprehensive overview of the organizational structure.

d. Employees:

Introducing the Employee Module, where simplicity meets efficiency. Easily access a curated list of employees who have undergone the seamless registration process on the home page. The Employee List feature ensures a quick and organized display of all registered employees.

e. Jobs:

After successful registration, organizations gain the ability to effortlessly create and manage jobs tailored to their needs. Dive into the customization process, specifying job details, descriptions, and the number of vacancies available. Define the required criteria with ease, empowering your organization to attract the perfect candidates.

f. Candidates:

The Candidate's Tab effortlessly navigates through applied, recommended, hired, declined, and withdrawn applicants, gaining comprehensive control over the recruitment pipeline. Elevate the communication game by seamlessly sending emails to selected candidates, notifying them about upcoming interviews. Take the hiring process to the next level by setting up convenient online interviews directly within the platform.

g. Statistics:

Unlock insights and make data-driven decisions with the Statistics Page! This dynamic feature calculates and presents a comprehensive overview of your recruitment process. Gain valuable insights into the total count of jobs created and applied candidates. Visualize the success with bar charts illustrating the number of candidates who applied and were hired.

h. Settings:

Take control of your account with our Settings Tab! Effortlessly edit the profile to reflect the latest information. Feel secure with the Change Password option, where a seamless process awaits. When changing the password, rest easy knowing that a verification email will be promptly sent to the registered email address, ensuring the utmost security. Enjoy a user-friendly experience as we can manage and personalize the account settings with ease through the intuitive Settings Tab.

2. APPLICANT MODULE :

a. Register:

Upon signing up for the first time, users will receive a verification email to confirm their registration. Clicking the verification link within the email will securely redirect them to the login page, allowing seamless access to their account. This process ensures the security and validity of user accounts while facilitating a smooth login experience.

b. Login :

The login page serves as the gateway to secure access, seamlessly verifying the authenticity of the entered email address and password. By conducting a thorough validation process, the system ensures that only valid email credentials gain entry. Once verified, users are granted access to the home page.

c. Home:

Upon logging into the platform's home page, users are greeted with a versatile interface that seamlessly integrates organizational management and job application functionalities. Users have the option to dynamically create their organizations, assuming the role of organizers with the ability to tailor and manage their organizational profiles.

d. View jobs:

Explore exciting career opportunities in the View Jobs section! Here, all jobs created by the organization are showcased for potential candidates. Users can easily apply for positions that align with their skills and interests by providing the necessary details as outlined in the job description. Streamline the application process by submitting resumes directly through the platform.

e. Apply jobs:

Once users select a job of interest from the "View Jobs" section, a streamlined application process awaits. Users are presented with a comprehensive job description, enabling them to make informed decisions about their application. To apply, users can effortlessly fill in their details through a user-friendly interface, users are also allowed to upload their resumes, allowing them to showcase their qualifications and experiences seamlessly.

f. Receive mail:

Once users have successfully applied for a job, the platform takes the next step in fostering clear communication and transparency throughout the hiring process. Applicants can expect to receive a prompt email containing essential details about their interview schedule, procedures, and additional information. Furthermore, in the event of successful selection, candidates are promptly notified through a confirmation email.

g. Settings:

Take control of your account with our Settings Tab! Effortlessly edit the profile to reflect the latest information. Feel secure with the Change Password option, where a seamless process awaits. When changing the password, rest easy knowing that a verification email will be promptly sent to the registered email address, ensuring the utmost security. Enjoy a user-friendly experience as we can manage and personalize the account settings with ease through the intuitive Settings Tab.

SYSTEM DESIGN

5. SYSTEM DESIGN

The design is the process of applying various techniques and principles. It is a process through which requirements are translated into a representation of software. System design involves translating information requirements and conceptual design into technical specifications and the general flow of processing. The proposed system consists of various modules, their maintenance works are finally report generation.

Various designing process including input design, output design, database design, etc., is involved in this phase. The overall efficiency of the system depends mainly on how this phase has been implemented.

The physical design relates to the actual input and output processes of the actual input and output processes of the system. This is explained in terms of how data is input into a system, how it is verified or authenticated, how it is processed, and how it is displayed. In physical design, the following requirements about the system are decided.

- Input requirement
- Output requirement

System design is the phase that bridges the gap between problem domain and the existing system in a manageable way. This phase focuses on the solution domain, i.e. “how to implement”. It is the phase where the SRS document is converted into a format that can be implemented and decides how the system will operate.

In this phase, the complex activity of system development is divided into several smaller sub-activities, which coordinate with each other to achieve the main objective of system development. System design includes the following. They are,

- Identify Design goals
- System Decomposition
- Identification of concurrency
- Hardware Allocation
- Data Management
- Global Resource Hand Leap
- Software Control implementation
- Boundary condition

5.1.INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

- Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
- It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

- When the data is entered it will check for its validity. Data can be entered with the help of screens.

5.2. OUTPUT DESIGN:

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively.
2. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
3. Select methods for presenting information.
4. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

5.3. DATABASE DESIGN:

A database is a collection of interrelated data items that are stored with minimum redundancy to serve many users quickly and efficiently. The proposed system maintains a database of all information related to users and staffs list, staff request form, crime data form.

Data integrity checks have been incorporated in all the tables to avoid duplication of data. Hence the tables should have primary key constraints for integrity check.

Organization:

Column	Type	Constraint
id	Int	Primary Key
FullName	varchar(50)	Not Null
AdminEmail	varchar(100)	Not Null
Password	varchar(50)	Not Null
UpdationDate	Timestamp	Not Null

Applicant:

Column	Type	Constraint
id	Int	Primary Key
UserID	varchar(50)	Not Null
Fullname	varchar(100)	Not Null
EmailId	varchar(100)	Not Null
MobileNumber	Int	Not Null
Password	varchar(10)	Not Null
Status	Int	Not Null
RegDate	Timestamp	Not Null
UpdationDate	Timestamp	Not Null

tblbooks:

Column	Type	Constraint
Id	Int	Primary Key
BookName	varchar(100)	Not Null
CatId	Int	Not Null
AuthorId	Int	Not Null
BkId	varchar(20)	Not Null
Bklocation	varchar(20)	Not Null
Bkimage	varchar(50)	Not Null
RegDate	Timestamp	Not Null
UpdationDate	Timestamp	Not Null
book_issue_status	enum('Return','Issue')	Not Null

tblcategory:

Column	Type	Constraint
Id	Int	Primary Key
CategoryName	varchar(50)	Not Null
Status	Int	Not Null
CreationDate	Timestamp	Not Null
UpdationDate	Timestamp	Not Null

tblauthor:

Column	Type	Constraint
Id	Int	Primary Key
AuthorName	varchar(50)	Not Null
Status	Int	Not Null
CreationDate	Timestamp	Not Null
UpdationDate	Timestamp	Not Null

tblissuedbooks:

Column	Type	Constraint
Id	Int	Primary Key
BookId	Int	Not Null
UserID	varchar(30)	Not Null
IssueDate	Date	Not Null
ReturnDate	Date	Not Null
Status	Int	Not Null
Fine	Int	Not Null
EmailId	varchar(60)	Not Null
Book_issue_status	Enum('Return','Issue')	Not Null

mail:

Column	Type	Constraint
Receiver	varchar(100)	Not Null
Body	varchar(500)	Not Null

feedback:

Column	Type	Constraint
ID	Int	Primary Key
UserID	varchar(30)	Not Null
Name	varchar(50)	Not Null
Email	varchar(100)	Not Null
Comment	varchar(100)	Not Null
Suggestion	varchar(500)	Not Null

5.4. DATA FLOW DIAGRAM

A data flow diagram is a graphical tool. The system models are termed as data flow diagrams (DFD). It is used to describe and analyse the movement of data through a system – manual or automated. This is a central tool and the basis from which the other component is developed.

The data flow diagram is based on the top-down approach in this system. A level 0 DFD, also called as context model, represents the entire software element as a single bubble with the input and output data indicated by incoming and outgoing arrows. The data flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart. There are several notations for displaying a data-flow diagram. The data-flow diagram is part of the structured analysis modeling tools.

DFD consists of processes, flows, warehouses, and terminators. The data flow diagram shows the transfer of information from one part of the system to another. The symbol of the flow is the arrow. It is determined for system developers, on the one hand, and project contractors on the other, so the entity names should be adapted for model domains or professionals. It is necessary to maintain consistency across all DFD levels. DFD must be consistent with other models of the system – ERD, STD, Data Dictionary, and Process Specification models.

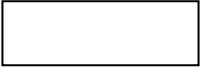
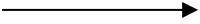
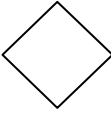

Creating a dataflow diagram will allow the programmer to create a program with minimal discomfort in programming the code and further increase the productivity of the programmer or program group. Data flow diagrams help the programmer figure out what options the programs will need in order to handle the data it is given.

Using the data flow diagram, it makes easy to explain the program to the people. It will definitely save the amount of time for programmer would have spent explaining the code to other people.

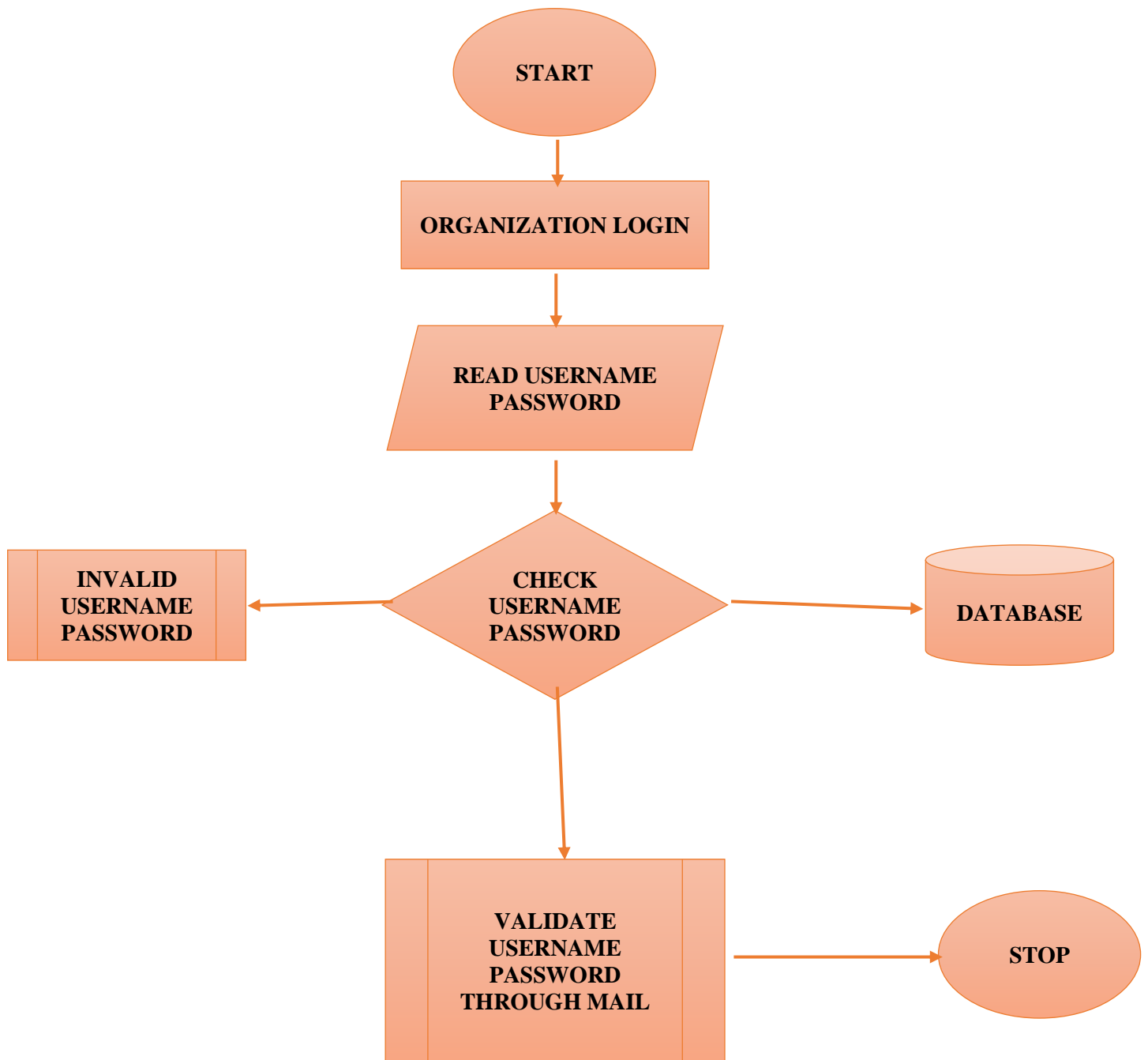
The data flow diagram will help the programmer be able to see what will happen if certain code is injected into the program. The debugging in data flow diagram instead of bigger coding will help the programmer to find the error and correct it.

DFD symbol

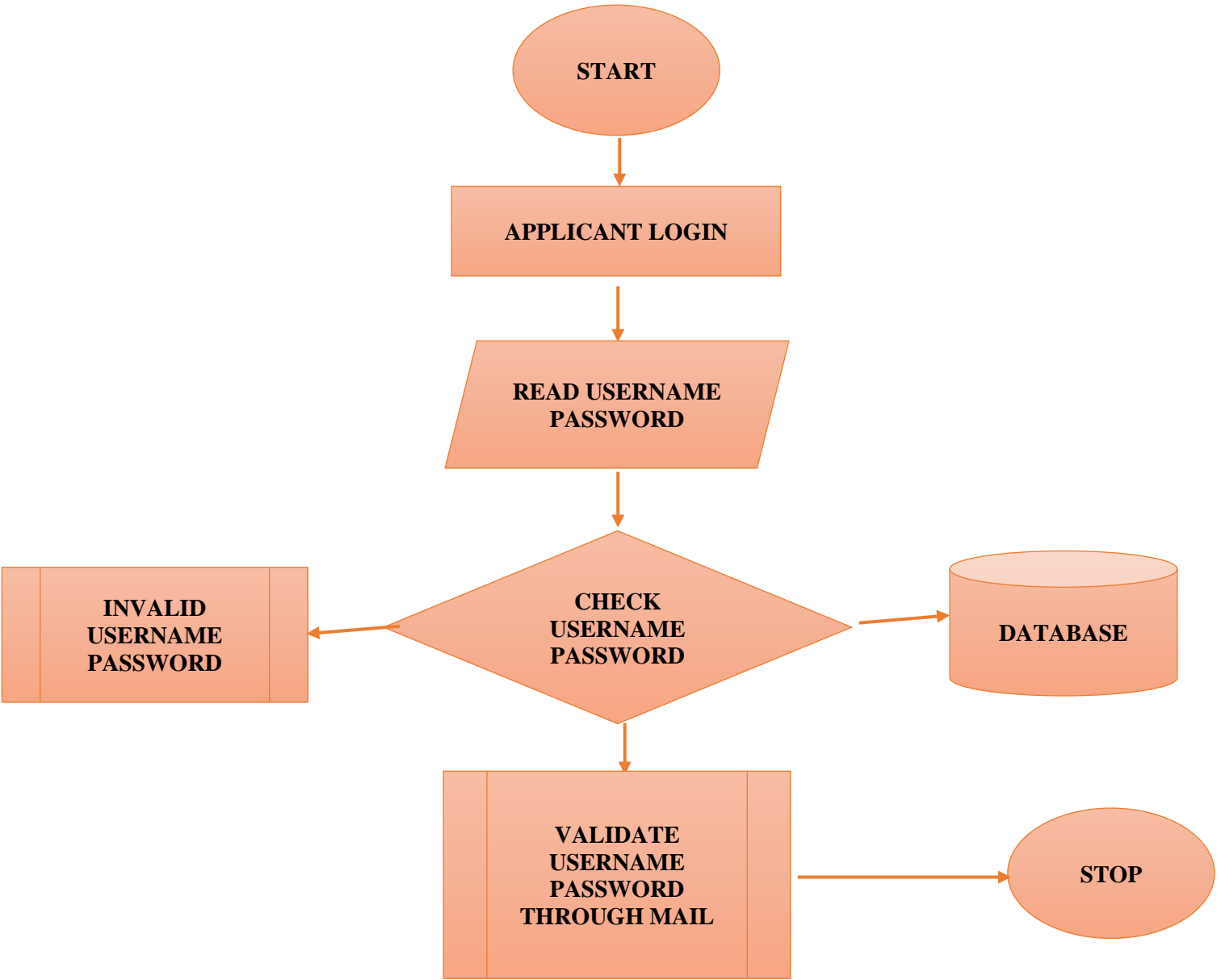
- A square defines a source or destination of system.
- An arrow identifies data flow (data motion).
- A diamond represents a condition or process that transforms incoming data flows into outgoing data flows.
- A rounded corner rectangle represents the process.

	Source or Destination of system
	Data Flow
	Conditions
	Process

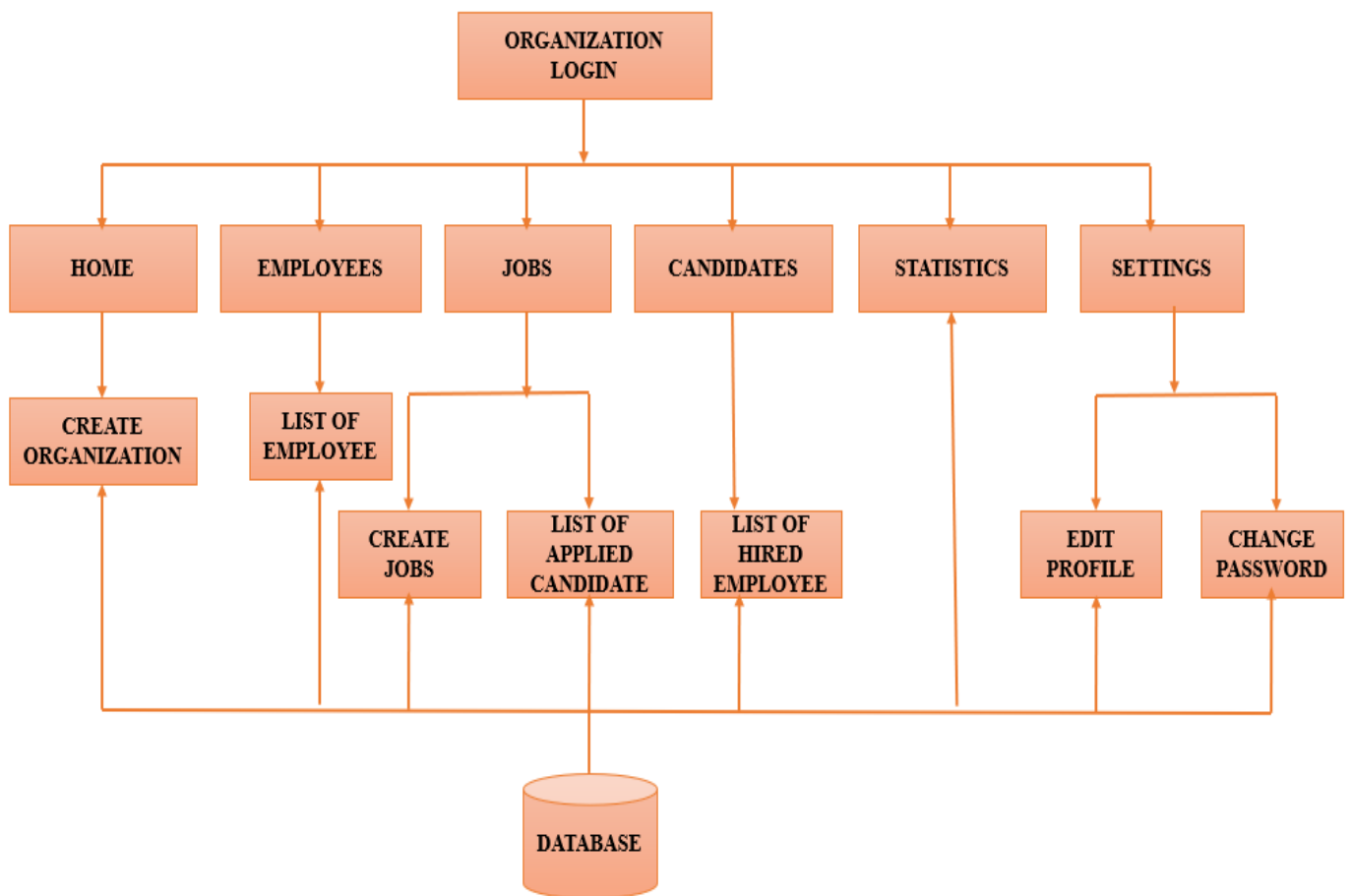
ORGANIZATION LOGIN:



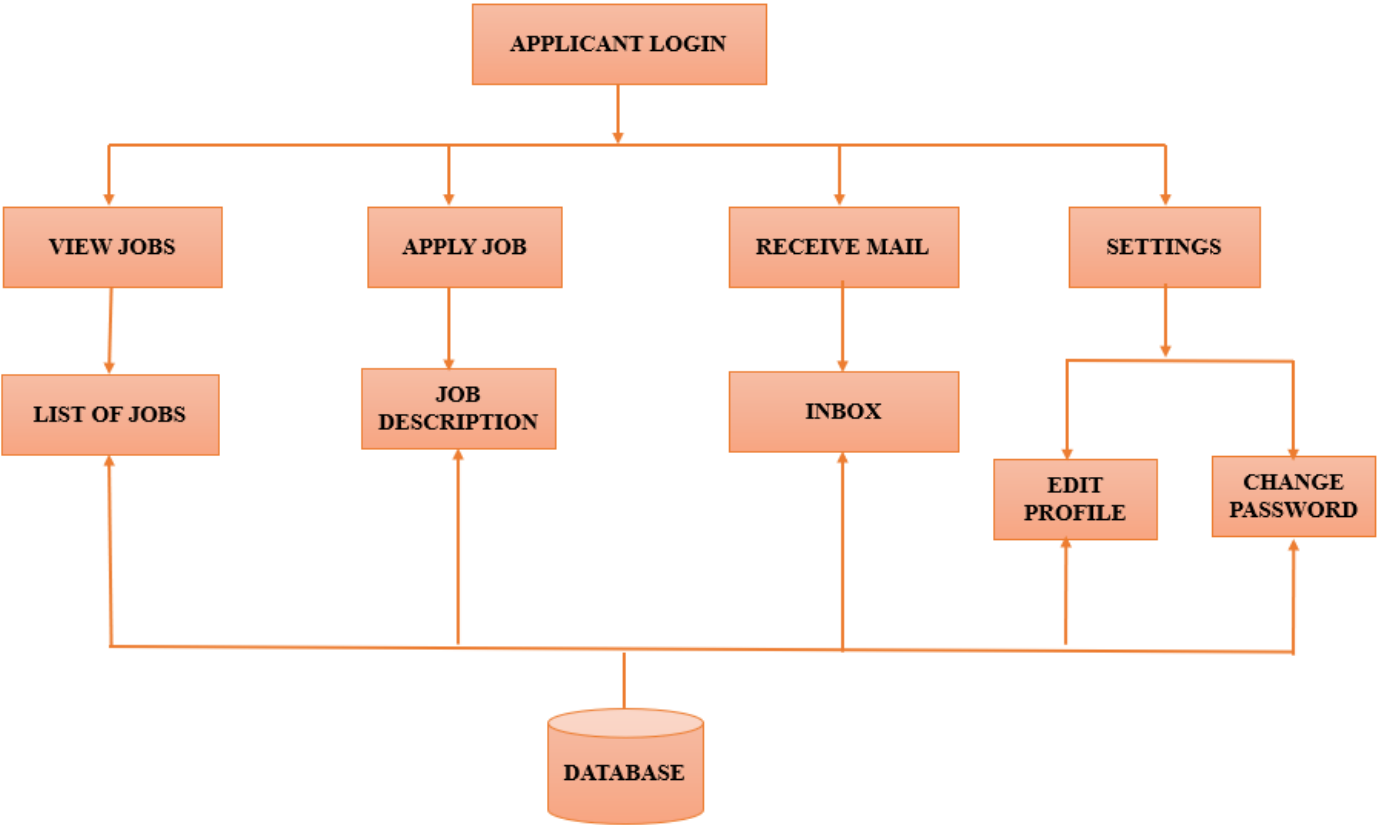
APPLICANT LOGIN:



ORGANIZATION:



APPLICANT:



SYSTEM TESTING & IMPLEMENTATION

6. SYSTEM TESTING & IMPLEMENTATION

6.1 SYSTEM TESTING

System Testing is an important stage in any system development life cycle. Testing is a process of executing a program with the intention of finding errors. The importance of software testing and its implications with respect to software quality cannot be overemphasized. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. A good test case is one that has a high probability of finding a yet undiscovered error.

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

Testing is the set of activities that can be planned in advance and conducted systematically. Different test conditions should be thoroughly checked and the bugs detected should be fixed.

The testing strategies formed by the user are performed to prove that the software is free and clear from errors. To do this, there are many ways of testing the system's reliability, completeness and maintainability.

The important phase of software development is concerned with translating the design specification into the error-free source code.

The system testing was carried out in a systematic manner with a test data containing all possible combinations of data to check the features of the system. A test data was prepared for each module, which took care of all the modules of the program.

System Testing is an important stage where the system developed is tested with duplicate or original data. It is a process of executing a program with the intent of finding an error. It is a critical process that can consume fifty percent of the development time.

System testing is performed on the entire system in the context of either functional requirement specifications (FRS) or system requirement specification (SRS), or both. System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software or hardware requirements specifications.

All the front-end and back-end connectivity are tested to be sure that the new system operates in full efficiency as stated. The purpose of the system testing is to consider all the likely variations to which it will be suggested and push the system to its limits. Testing is done for each module.

After testing all the modules, the modules are integrated and testing of the final system is done with the test data, specially designed to show that the system will operate successfully in all its aspects conditions. The testing process focuses on logical intervals of the software ensuring that all the statements have been tested and on the function intervals.

The following are the attributes of good test:

- A good test is not redundant.
- A good test should be "best of breed".
- A good test should be neither simple nor too complex.

6.1.1.UNIT TESTING

In the unit testing the analyst tests the program making up a system. The software units in a system are the modules and routines that are assembled and integrated to perform a specific function. In a large system, many modules on different levels are needed.

Unit testing can be performed from the bottom up starting with the smallest and lowest level modules and proceeding one at a time. For each module in a bottom-up testing, a short program executes the module and provides the needed data.

ADVANTAGES:

- It will find problems early in the development cycle.
- This includes both bugs in the programmer's implementation and flaws or missing parts of the specification for the unit.
- Unit testing allows the programmer to refactor code at a later date, and make sure the module still works correctly.
- Due to the modular nature of the unit testing, we can test parts of the project without waiting for others to be completed.

DISADVANTAGES:

- Unit testing can't be expected to catch every error in a program. It is not possible to evaluate all execution paths even in the most trivial programs.
- Unit testing by its very nature focuses on a unit of code. Hence it can't catch integration error or broad system level errors.

6.1.2.INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure while conducting test to uncover errors associate with interfacing. Objectives are used to take unit test modules and built program structure that has been directed by design.

The integration testing is performed for this project when all the modules where to make it a complete system. After integration the project works successfully.

The integration testing can be carried out using two methodologies:

- Top Down Integration
- Bottom Up Integration

Top Down is an approach to integrated testing where the top integrated modules are tested and the branch of the module is tested step by step until the end of the related module.

Bottom up testing is an approach to integrated testing where the lowest level components are tested first, then used to facilitate the testing of higher level components.

The process is repeated until the component at the top of the hierarchy is tested. Hence, the objective of the integration testing is to take unit.

6.1.3.VALIDATION TESTING

Validation testing can be defined in many ways, but a simple definition is that can be reasonably expected by the customer. After validation test has been conducted, one of two possible conditions exists.

- The functions or performance characteristics confirm to specification and are accepted.
- A deviation from specification is uncovered and a deficiency list is created.

Proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. For example, in this project validation testing is performed against module. This module is tested with the following valid and invalid inputs for the field id.

6.1.4.WHITE BOX TESTING

White box testing, sometimes called glass-box testing is a test case design method that uses the control structure of the procedural design to derive test cases. Using white box testing methods, the software engineer can derive test cases that

- Guarantee that all independent paths with in a module have been exercised at least once.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds and
- Exercise internal data structure to assure their validity.

For example, in this project white box testing is performed against patient module.

Without entering text if we apply it displays the message “First add record then save it” else it should be saved.

6.1.5.BLACK BOX TESTING

This method treats the coded module as a black box. The module runs with inputs that are likely to cause errors. Then the output is checked to see if any error occurred. This method cannot be used to test all errors, because some errors may depend on the code or algorithm used to implement the module.

6.2 IMPLEMENTATION

Implementation is the stage in the project where the theoretical design is turned into a working system. The most critical stage is achieving a successful system and in giving confidence on the new system for the users, what it will work efficient and effectively.

It involves careful planning, investing of the current system, and its constraints on implementation, design of methods to achieve the changeover methods.

The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out in these plans; discussion has been made regarding the equipment, resources and how to test activities.

The coding step translates a detail design representation into a programming language realization. Programming languages are vehicles for communication between human and computers programming language characteristics and coding style can profoundly affect software quality and maintainability. The coding is done with the following characteristics in mind.

- Ease of design to code translation
- Code efficiency
- Memory efficiency
- Maintainability

Implementation is the stage of the project when the theoretical design is turned out into a working system.

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

6.2.1 IMPLEMENTATION PLAN

It is the phase that focuses on user training, site preparation and file conversion. The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that will work efficiently and effectively. The system can be implemented only after thorough testing is done and found to work according to the specification. Two major tasks of preparing the implementation are education, training of the users and testing the systems. System analysis and design efforts will be more for complex systems being implemented.

- The implementation plan includes:
- The role of the implementing agency
- Beneficiary participation
- Organizational structure and staffing
- Sustainability
- Reporting System
- Time control and remedial action
- Schedule and supervision
- Financial Management

6.2.2 IMPLEMENTATION PROCESS

The implementation process begins with preparing a plan for the implementation system. According to this plan, the other activities are to be carried out. The success of a system depends upon how they are operated and used. Thus the quality of training is connected to the success of the system.

SAMPLES

7. SAMPLES

7.1 CODING:

Client

App.js

```
import { configureStore } from '@reduxjs/toolkit'

import OrganizationDetailsReducer from '../Features/Dashboard/Organization_Details_Slice'

import OrganizationDetails from '../Features/JobCycle/storeAllCandidatesDetails'

export const store = configureStore({

  reducer: { OrganizationDetailsReducer, }

})
```

Organization_Details_Slice.js

```
import { createSlice } from '@reduxjs/toolkit'

const initialState = {

  apiData: null,

  loading: false,

  error: null,

}
```

```
export const OrganizationDetails = createSlice({

  name: 'organization_details',

  initialState,

  reducers: {

    fetchOrganizationDataStart(state) {

      state.loading = true;

      state.error = null;

    },

    fetchOrganizationDataSuccess(state, action) {

      state.loading = false;

      state.apiData = action.payload;

    },

    fetchOrganizationDataFailure(state, action) {

      state.loading = false;

      state.error = action.payload;

    },

  },

},
```

```
export const { fetchOrganizationDataStart, fetchOrganizationDataSuccess, fetchOrganizationDataFailure } = OrganizationDetails.actions
```

```
export default OrganizationDetails.reducer
```

storeAllCandidatesDetails.js

```
import { createSlice } from "@reduxjs/toolkit";
```

```
const initialState = {
```

```
  candidatesData: null,
```

```
  loading: false,
```

```
  error: null,
```

```
  filterBS: null
```

```
}
```

```
export const storeAllCandidatesDetails = createSlice({
```

```
  name: 'StoreCandidatesDetails',
```

```
  initialState,
```

```
  reducers: {
```

```
    startFetchingCandidatesData(state) {
```

```
      state.loading = true;
```



```

    },

    sucessOnFetchingCandidatesData(state, action) {

        state.loading = false;

        state.canidatesData = action.payload

    },

    errorFetchingCandidatesData(state, action) {

        state.loading = false,

        state.error = action.payload;

    },

    filterOnBS(state, action) {

        return state.filterBS = "Hamza"

    }

}

}))

export const { filterOnBS, startFetchingCandidatesData, sucessOnFetchingCandidatesData,
errorFetchingCandidatesData } = storeAllCandidatesDetails.actions;

export default storeAllCandidatesDetails.reducer;

```

CreateJobHeader.jsx

```
import axios from "axios";

import React from "react";

import { useState } from "react";

import { Link } from "react-router-dom";

import DownImg from "../../assets/icons/down.svg";

function CreateJobHeader({ setData }) {

  const [jobStatus, SetJobStatus] = useState(false);

  const [departmentStatus, SetDepartmentStatus] = useState(false);

  const filterShowClosedJobs = () => {

    // axios POST request

    const options = {

      url: "http://localhost:8080/job/get-jobs/closed",

      method: "POST",

      headers: {

        Accept: "application/json",

        "Content-Type": "application/json;charset=UTF-8",
```

```

    data: { id: localStorage.getItem("organization_id") },

};

axios(options).then((response) => {

    // console.log(response);

    setData(response.data.jobs);

});

};

const filterShowActiveJobs = () => {

    // axios POST request

    const options = {

        url: "http://localhost:8080/job/get-jobs/active",

        method: "POST",

        headers: {

            Accept: "application/json",

            "Content-Type": "application/json;charset=UTF-8",

        },

        data: { id: localStorage.getItem("organization_id") },

```

```

axios(options).then((response) => {

  // console.log(response);

  setData(response.data.jobs);

});

};

return (

  <div className="flex w-10/12 m-auto justify-center text-center items-center
topNavigationBarShadow bg-transparent mt-2 p-10 ml-12 h-14 ">

    { /* --> Main Create Job Button */}

    <div className="w-full sm:w-1/2 text-center">

      <Link to={"/postjob"}>

        <button

          type="submit"

          className="btnfont btn btn-md bg-primary border-none hover:bg-black"

        >

          Create New Job

```

</Link>

</div>

{/* 2nd flex div */}

<div className="w-full flex justify-end items-center -mr-10 sm:mr-12 ">

{/* ==> Job Status Button */}

<button

onClick={() => SetJobStatus(!jobStatus)}

className="btn bg-transparent text-secondary normal-case gap-2 ml-8 rounded-lg border-0 border-solid border-secondary hover:bg-primary hover:border-solid hover:border-primary hover:text-white "

>

Job Status

</button>

{jobStatus == true ? (

<div className="top-36 right-96 absolute dropdown-bottom">

<ul

tabIndex={0}

>

<li onClick={filterShowActiveJobs}>

<a>Active

<li onClick={filterShowClosedJobs}>

<a>Closed

</div>

): null}

{/* <button

onClick={() => SetDepartmentStatus(!departmentStatus)}

className="btn bg-transparent text-secondary normal-case gap-2 ml-8 rounded-lg border-0 border-solid border-secondary hover:bg-primary hover:border-solid hover:border-primary hover:text-white "

>

Department

```
</button> */}
```

```
{/* {departmentStatus == true ? (
```

```
<div className="top-36 absolute dropdown-bottom">
```

```
<ul
```

```
  tabIndex={0}
```

```
  className="dropdown-content menu p-2 shadow bg-base-100 rounded-box w-40 "
```

```
>
```

```
<li>
```

```
<a>IT</a>
```

```
</li>
```

```
<li>
```

```
<a>HR</a>
```

```
</li>
```

```
<li>
```

```
<a>Marketing</a>
```

```
</li>
```

```
</ul>
```

```
) : null} */}
```

```
</div>
```

```
</div>
```

```
);
```

```
}
```

```
export default CreateJobHeadar;
```

CreatedJobElement.jsx

```
import axios from "axios";
```

```
import React, { useEffect } from "react";
```

```
import { useState } from "react";
```

```
import { Link, useNavigate } from "react-router-dom";
```

```
import DeleteIcon from "../../assets/icons/delete.svg";
```

```
import SocialIcon from "../../assets/icons/share.svg";
```

```
function CreatedJobElement({ data, setData }) {
```

```
  const navigate = useNavigate();
```

```
  const handleJob = (id) => {
```

```
    navigate(`/JobDetails/${id}`);
```



```

// console.log(data);

return (

<div className="flex flex-wrap gap-6">

  {data?.map((e, index) => {

    return (

      <div

        key={index}

        onClick={(event) => handleJob(e._id)}

        title="Job"

        className="bg-white hover:bg-gray-100 hover:border hover:border-solid hover:border-gray-300
flex flex-wrap items-center w-80 pl-4 pr-4 pt-2 modalShadow cursor-pointer "

      >

        {/* <Link to={"/JobDetails"}> */}

        <div className=" w-full p-2 flex justify-between items-center ">

          <h2 className="heading3 inline font-medium">{e.jobPosition}</h2>

          <button

            className="inline float-right mr-4 p-2 w-20 rounded-full font-medium text-primarytext

```

>

Active

</button>

</div>

{/* PART TO HANDLE DATA */}

<div className="w-full flex h-28">

<div className="w-1/2 flex flex-col justify-center text-center">

<div className="heading4">Totall Candidates</div>

<div className="heading4 font-medium">{e.applicants_no}</div>

</div>

<div className="w-1/2 flex flex-col justify-center text-center">

<div className="heading4 ">Active Candidates</div>

<div className="heading4 font-medium">0</div>

</div>

</div>

```
<div className="flex flex-row w-full pb-4">
```

```
<div className=" w-4/5 ml-4">
```

```
<p className="text-sm heading4">JOB-ID: {index}</p>
```

```
</div>
```

```
<div className="flex justify-around items-center w-1/4">
```

```
<img src={DeleteIcon} alt="" className="inline w-4 h-4" />
```

```
<img src={SocialIcon} alt="" className="inline w-4 h-4" />
```

```
</div>
```

```
</div>
```

```
{/* </Link> */}
```

```
</div>
```

```
);
```

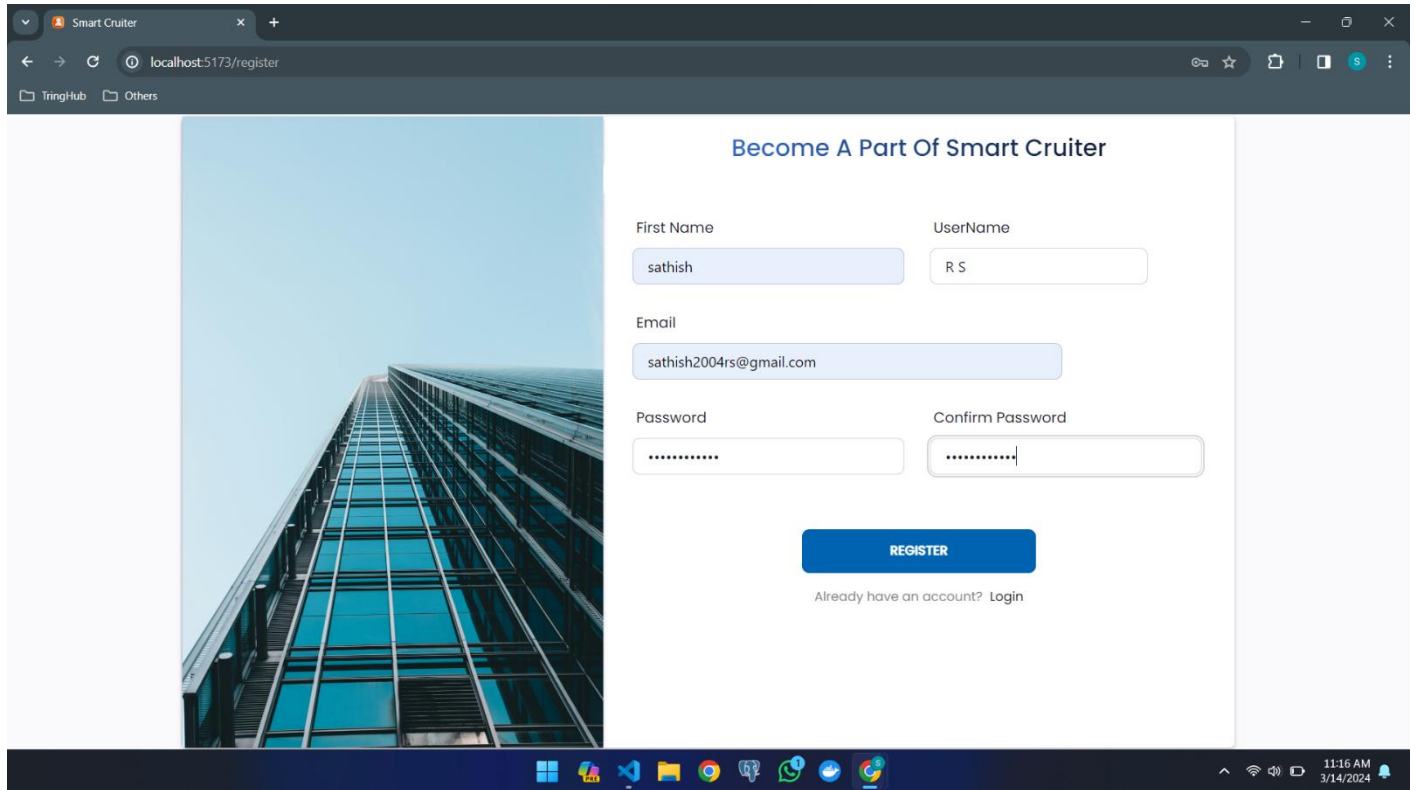
```
}}}
```

```
</div>
```

```
);
```

```
}
```

7.2 SCREENSHOTS:



The screenshot displays a web browser window with the title 'Smart Cruiser' and the address bar showing 'localhost:5173/register'. The page features a large background image of a modern glass skyscraper. On the right side, there is a registration form titled 'Become A Part Of Smart Cruiser'. The form includes input fields for 'First Name' (containing 'sathish'), 'UserName' (containing 'R S'), 'Email' (containing 'sathish2004rs@gmail.com'), 'Password', and 'Confirm Password'. A blue 'REGISTER' button is positioned below the password fields. At the bottom of the form, there is a link that says 'Already have an account? Login'. The Windows taskbar is visible at the bottom of the screen, showing various application icons and the system clock indicating 11:16 AM on 3/14/2024.

Become A Part Of Smart Cruiser

First Name

UserName

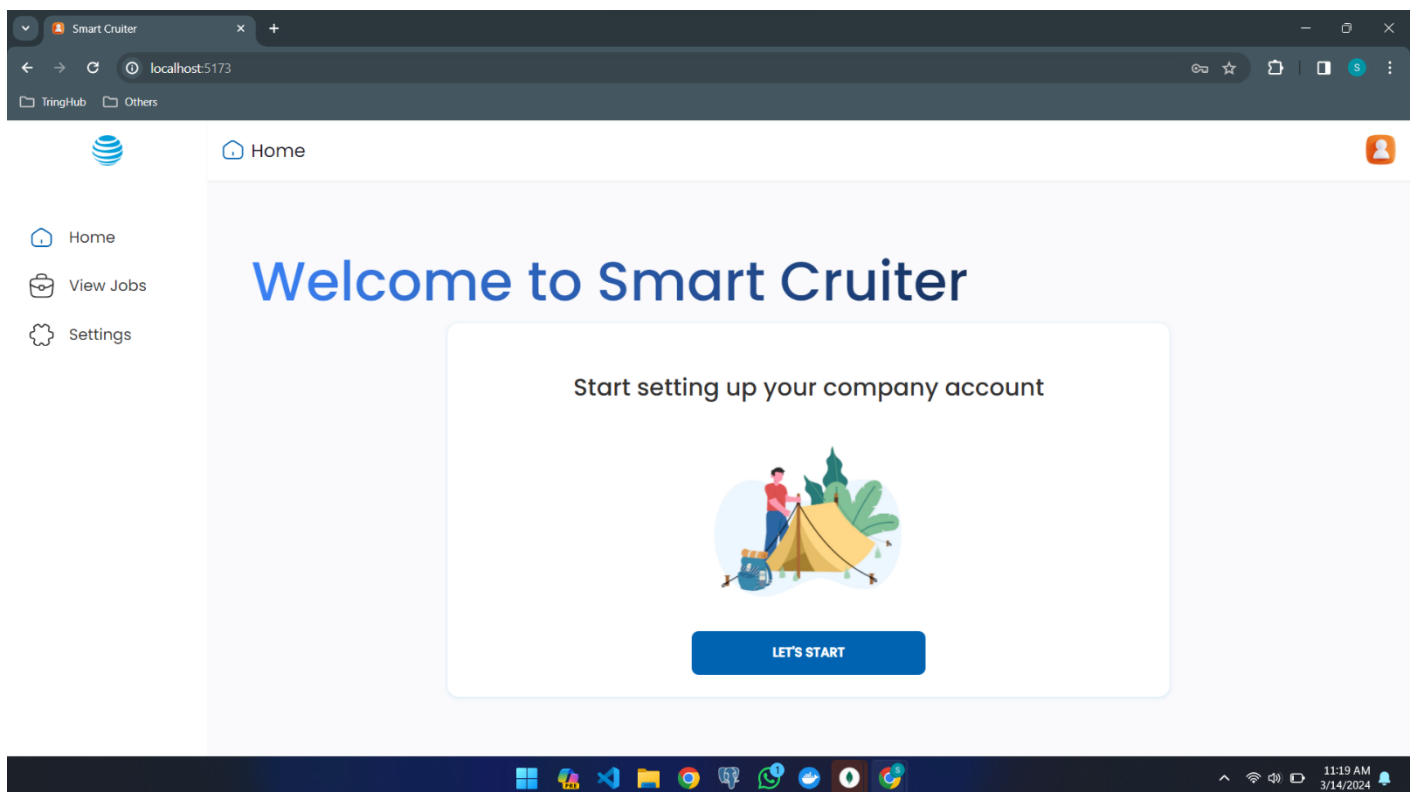
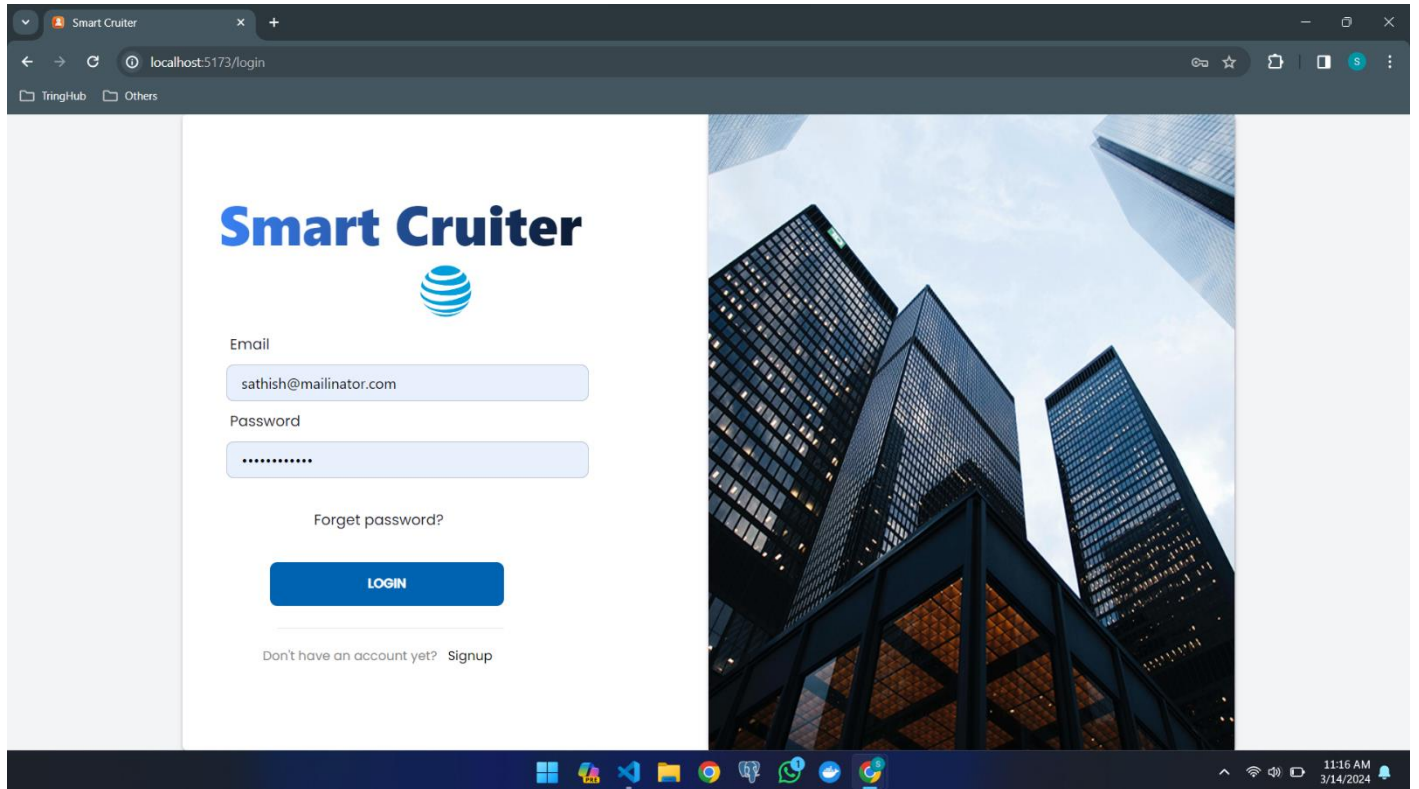
Email

Password

Confirm Password

REGISTER

[Already have an account? Login](#)



Product Manager

Fill out the form

Personal Information

First Name

Last Name

Date Of Birth

Gender

Address

City

Zip-Code



Click here to Upload profile picture

Smart Cruiser




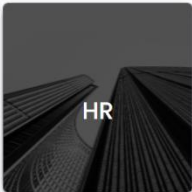
localhost:5173

TringHub Others

Home

TringApps

- Home
- Employees
- Candidates
- Jobs
- Statistics
- Settings



11:24 AM
3/14/2024

Smart Cruiser

localhost:5173/employees

TringHubOthers

Employees

Home

Employees

Candidates







Jobs

Statistics

Settings

Your Employees

ADD EMPLOYEES

NAME	EMAIL	ROLE	EDUCATION	SKILLS	EXPERIENCE	PERFORMANCE
 one	one@gmail.com					
 two	two@gmail.com					
 three	three@gmail.com					
 four	four@gmail.com					
 five	five@gmail.com	five				
 six	six@gmail.com	six	six	six	six	Average

11:24 AM

3/14/2024

Smart Cruiser

localhost:5173/Candidates

TringHubOthers

Candidates

Home

Employees

Candidates

Jobs


Statistics

Settings

Hired Candidate Details

Developer

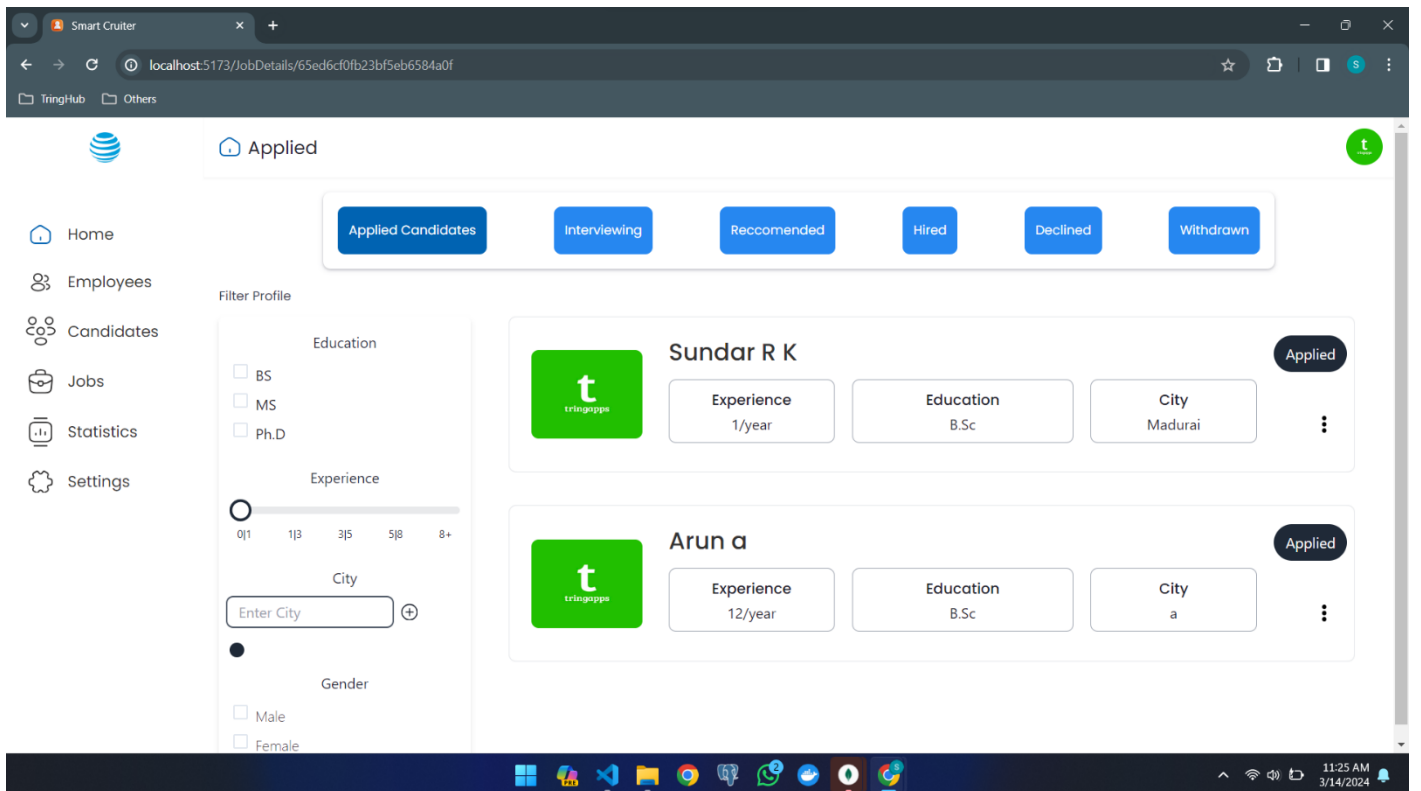
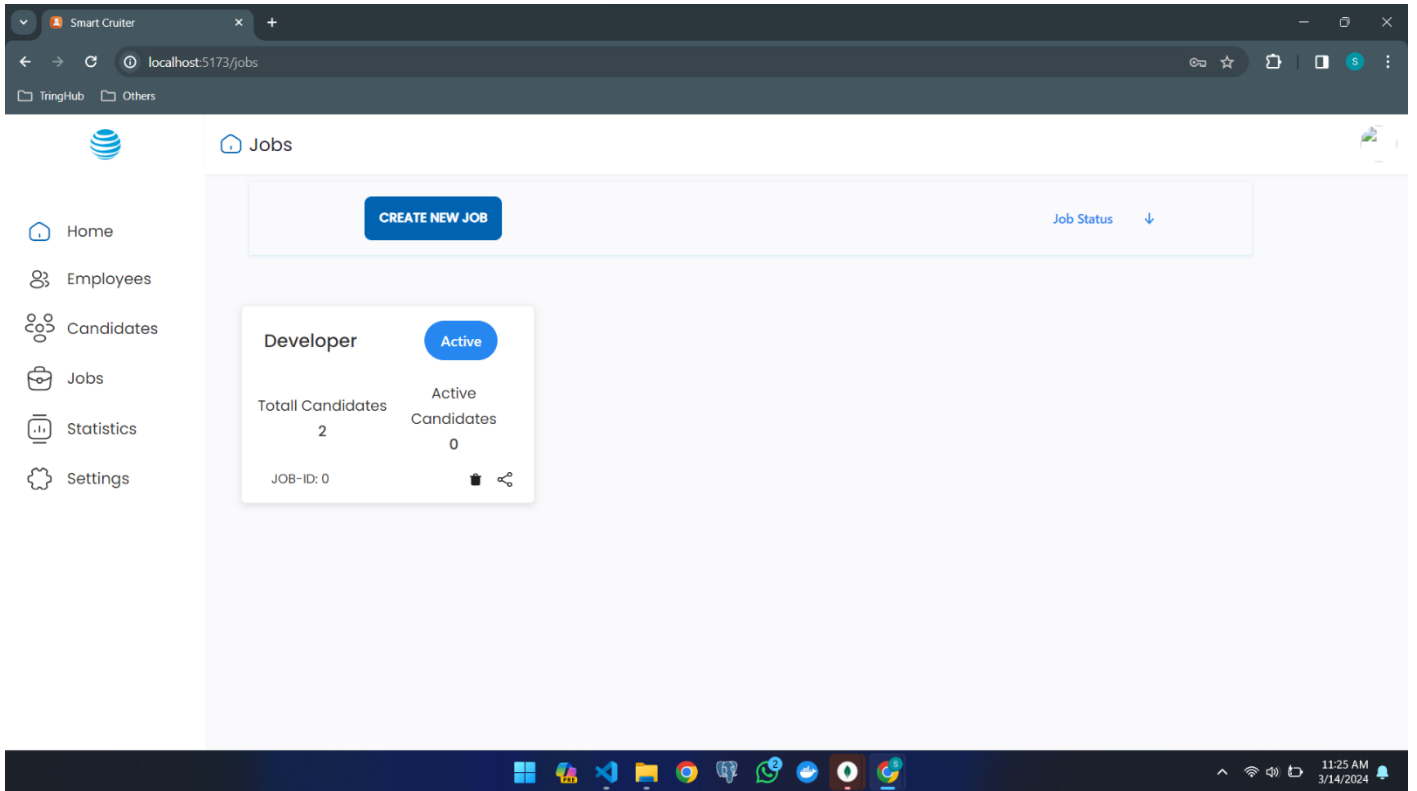
Frontend



2 Applied

Snipping Tool

Screenshot copied to clipboard and saved
Select here to mark up and share the image



Smart Cruiter

localhost:5173/JobDetails/applied/65ed707afb23bf5eb6584a59


TringHub

Others

JOB

Statistics

Settings



Sundar R K

DoB: 1/6/2023

Email: sathishrs2004@gmail.com

Contact: 1234567890

Address: ABC

VIEW RESUME


Education


Degree	Institute	Graduated In
Computer Engineering	Government College University, Faisalabad	2024

Experience

Title	Organization	Duration
Developer	Meta	1 Years

Social Handles


[LinkedIn](#)


[GitHub](#)

11:25 AM 3/14/2024

Smart Cruiter

localhost:5173/report

TringHub

Others

Home


Employees

Candidates

Jobs

Statistics

Settings



Statistics

Jobs Created

Till now


1

Applied Candidates

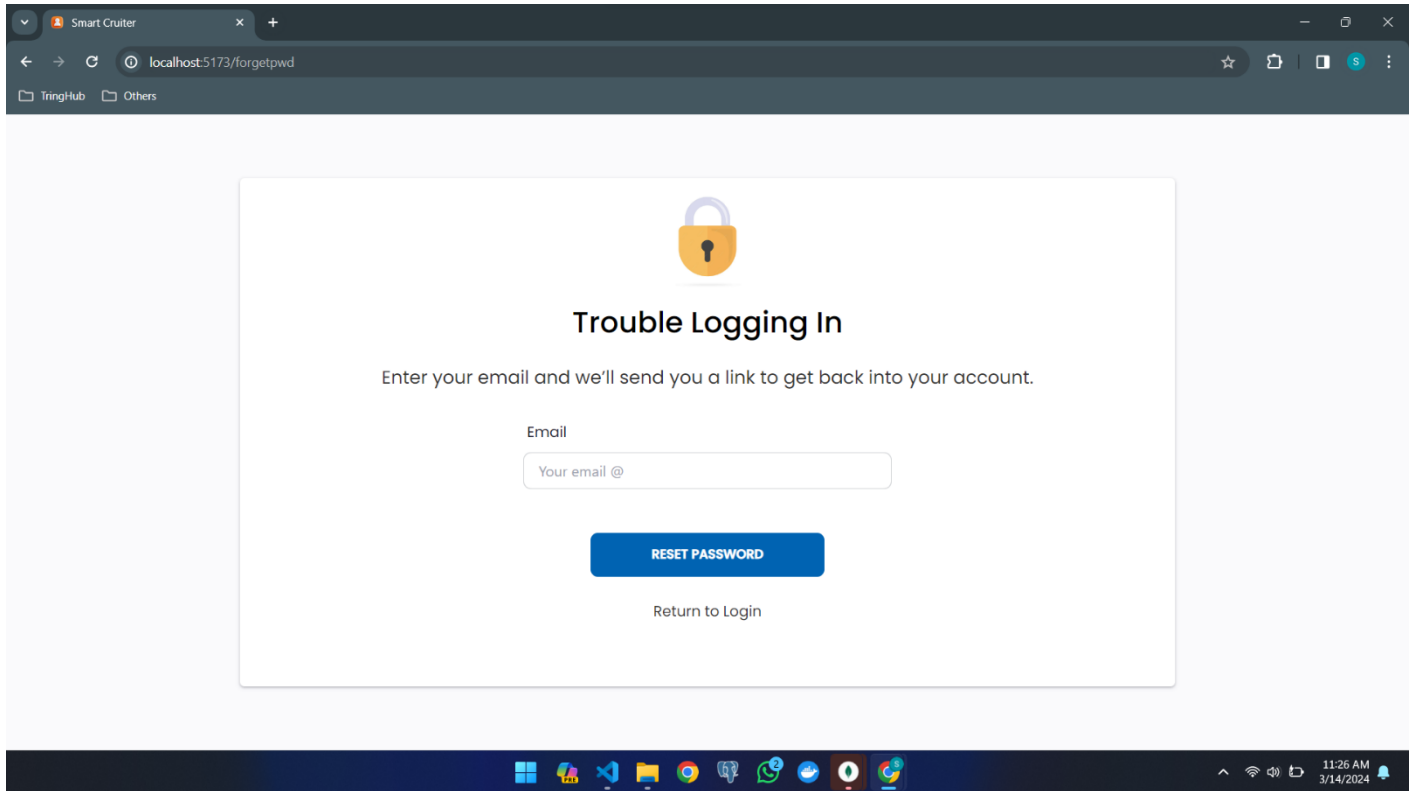
Totall

2

No of candidate applied and get hired



11:26 AM 3/14/2024



CONCLUSION

8. CONCLUSION

The Smart Recruiter project offers a comprehensive solution to modernize and streamline the recruitment process, leveraging cutting-edge technologies such as artificial intelligence and data analytics. Through its intuitive interface and advanced algorithms, it significantly enhances efficiency, accuracy, and effectiveness in sourcing and selecting top talent.

In conclusion, the Smart Recruiter project stands at the forefront of revolutionizing HR practices by providing a scalable, adaptable, and intelligent platform that caters to the evolving needs of businesses in today's competitive landscape.

By harnessing the power of technology, it empowers recruiters to make data-driven decisions, optimize resource allocation, and ultimately build high-performing teams. As organizations continue to prioritize talent acquisition and retention, the Smart Recruiter project emerges as a vital tool in driving success and achieving strategic objectives in the realm of human capital management.

FUTURE ENHANCEMENT

9. FUTURE ENHANCEMENT

Implement advanced natural language processing (NLP) and machine learning techniques to improve candidate matching algorithms. This could involve analyzing not only resumes but also social media profiles, professional portfolios, and other online presence to provide more accurate candidate recommendations.

Integrate predictive analytics to forecast future hiring needs based on historical data, market trends, and organizational growth projections. This could enable proactive talent acquisition strategies and mitigate talent shortages before they occur.

Develop features to support diversity and inclusion initiatives by providing insights into the diversity profile of candidate pools and identifying areas for improvement. This could involve incorporating bias detection algorithms and offering guidance on inclusive language in job postings.

BIBLIOGRAPHY

10. BIBLIOGRAPHY

10.1 BOOK REFERENCE

- State of MongoDB March, 2010. DB-Engines. Archived from the original on September 18, 2017. Retrieved July 5, 2017.
- Vaughan-Nichols, Steven J. "MongoDB "open-source" Server Side Public License rejected". ZDNet. Archived from the original on January 16, 2019. Retrieved January 17, 2019.
- Krebs, Brian. "Extortionists Wipe Thousands of Databases, Victims Who Pay Up Get Stiffed". krebsonsecurity.com. Brian Krebs. Archived from the original on January 11, 2017. Retrieved January 11, 2017.
- Betz, Brandy (October 30, 2019). "MongoDB teams with Alibaba Cloud". Seeking Alpha. Retrieved October 31, 2019.

10.2 WEB REFERENCE

- <https://codecourse.com/subjects/php>
- <https://www.tutorialspoint.com/php/>
- <https://www.w3schools.com/php/default.asp>
- <https://stackoverflow.com/>
- <https://www.codecademy.com/>
- <https://www.mongodb.com/>
- <https://www.w3schools.com/mongodb/>
- <https://react.dev/>
- <https://www.w3schools.com/REACT/DEFAULT.ASP>