PROFESSIONAL TRAINING REPORT

At

Sathyabama Institute of Science and Technology

(Deemed to be University)

Submitted in partial fulfillment of the requirements for

the award of Bachelor of Engineering Degree in

Computer Science and Engineering

By

P. JYOSHMITHA REDDY
REG. NO: 39110419

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI

CHENNAI– 600119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **P. JYOSHMITHA REDDY (Reg. No: 39110419)** who carried out the project entitled "**Employee Management System**" under my supervision from Dec 2022 to Apr 2022.

Internal Guide

Dr. N. Srinivasan

Head of the Department

_____

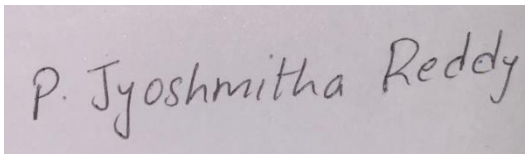
Submitted for Viva voce Examination held on_____

Internal Examiner **External Examiner**

DECLARATION

I, **P. JYOSHMITHA REDDY** hereby declare that the project report entitled **Employee Management System** done by me under the guidance of **Dr. N. Srinivasan** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE: 10/04/2022 **PLACE: CHENNAI**



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D**, **Dean**, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of **Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. N. Srinivasan** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE

TABLE OF CONTENTS

ABSTRACT

Employees are the backbone of any company therefore their management plays a major role in deciding the success of an organization. Employees Management Software makes it easy for the employer to keep track of all records. This software allows the administrator to edit employees, add new employees, transfer/promote/terminate employees. Each employee in the database is associated with a position can be added and edited when need arises. Employees can be transferred between positions easily without having to retype back their information in the database. You can check to see if there are duplicate positions/employees in the database. Most of all, the employer can assign tasks to employees and assess their progress in order to keep track of employee performance. A flexible and easy to use Employee Management software solution for small and medium sized companies provides modules for personnel information management thereby organization and companies are able to manage the crucial organization asset – people. The combination of these modules into one application assures the perfect platform for re-engineering and aligning Human Resource processes along with the organizational goals. This system brings about an easy way of maintaining the details of employees working in any organization. It is simple to understand and can be used by anyone who is not even familiar with simple employees system. It is user friendly and just asks the user to follow step by step operations by giving easy to follow options. It is fast and can perform many operations for a company. The goal of this project is to design and develop an employee management system to fill existing gaps in the electronic management of employees.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

EMS – Employee Management System

MSS – Management Self-Service

HRMS – Human Resource Management System

HRIS – Human Resource Information System

HR – Human Resource

HOD – Head of Department

ESS – Employee Self-Service

WBS – Work Breakdown Structure

ERP – Enterprise Resource Planning

CHAPTER 1

1. INTRODUCTION:

1.1 EMPLOYEE MANAGEMENT SYSTEM

Employees are the backbone of any company therefore their management plays a major role in deciding the success of an organization. Employees Management Software makes it easy for the employer to keep track of all records. This software allows the administrator to edit employees, add new employees, transfer/promote/terminate employees. Each employee in the database is associated with a position can be added and edited when need arises. Employees can be transferred between positions easily without having to retype back their information in the database. You can check to see if there are duplicate positions/employees in the database. Most of all, the employer can assign tasks to employees and assess their progress in order to keep track of employee performance.

Most of the contemporary Information systems are based on the Database technology as a collection of logically related data, and DBMS as a software system allowing the users to define, create, maintain and control access to the database. The process of constructing such kind of systems is not so simple. It involves a mutual development of application program and database. The application program is actually the bridge between the users and the database, where the data is stored. Thus, the well-developed application program and database are very important for the reliability, flexibility and functionality of the system. The so defined systems differentiate to each other and their development comprises a great variety of tasks to be resolved and implemented. The basic idea can be depicted on Figure 1.1 below:

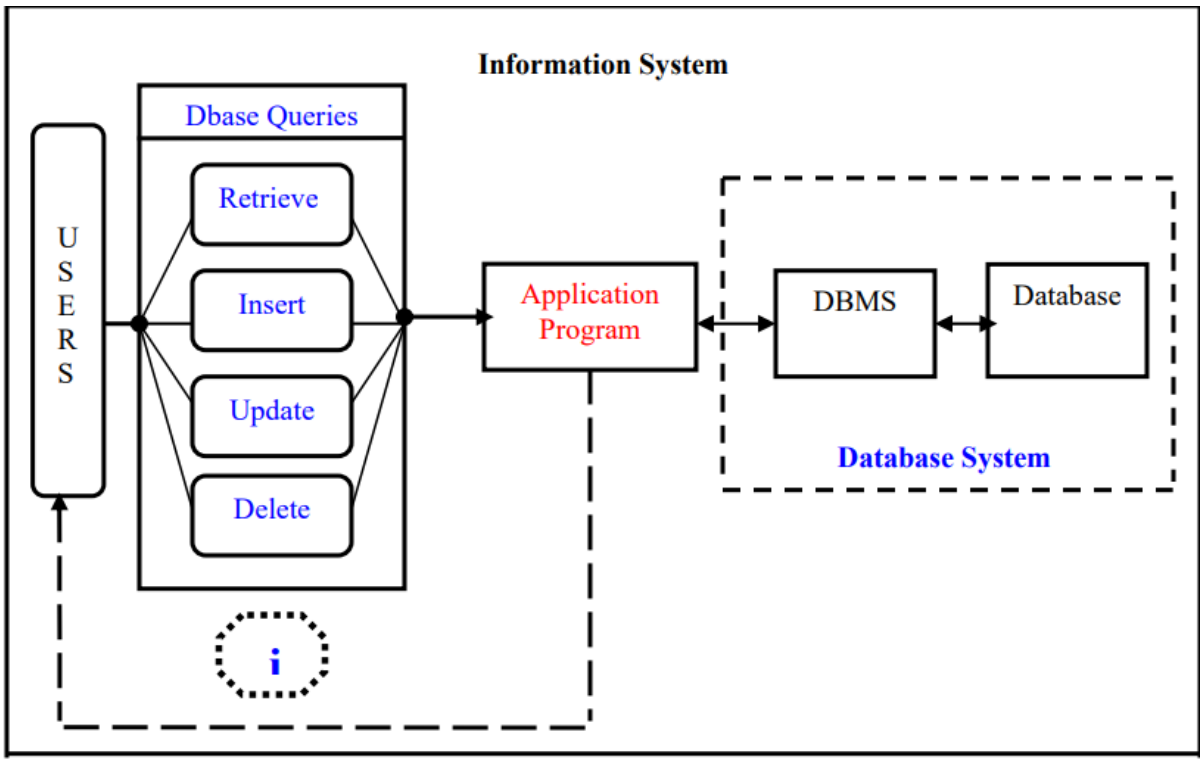


Figure 1.1 Database information systems - principle scheme

Information system suggests a computer technology to be used in order to provide information to users in an organization (for instance), as for the purposes of data transformation into useful information; computer hardware and software are designed and used.

1.2 METHOD

At the very commencement, I proceeded to a decision to carry out the development of my task into the following steps:

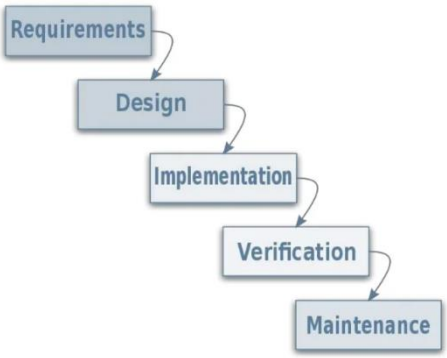
1. Exploring the available development environments and techniques.
2. Database Analyzing.
3. Database design and Implementation.
4. Program's Structure Analyzing.
5. GUI (Graphical User Interface) constructing.
6. Bringing all the stuff together (controls data binding and functions implementation).
7. Tests.

1.3 TECNOLOGIES USED

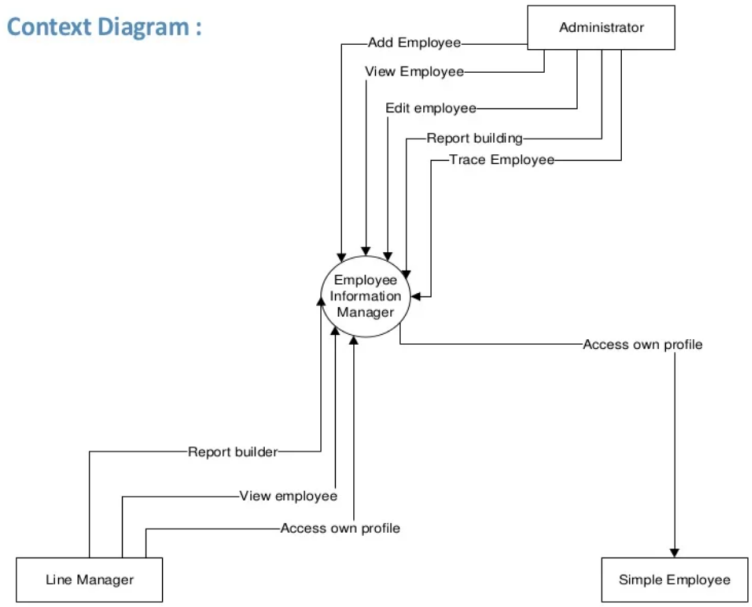
- HTML
- CSS
- JAVASCRIPT
- MONGODB
- EXPRESS
- EJS TEMPLATE
- NODE JS
- JSON
- ETC.

1.4 FLOW DIAGRAM

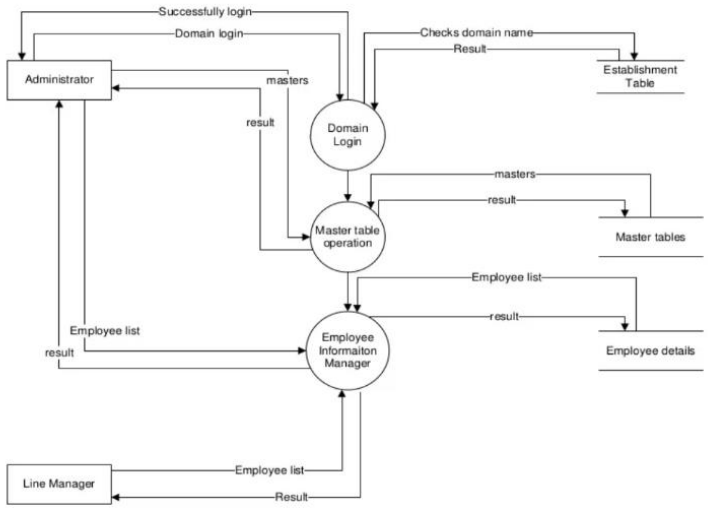
WATERFALL MODEL:



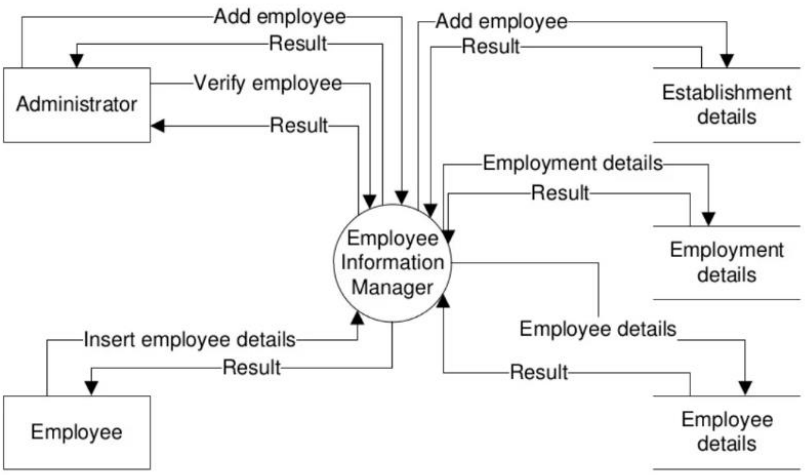
DATA FLOW DIAGRAM



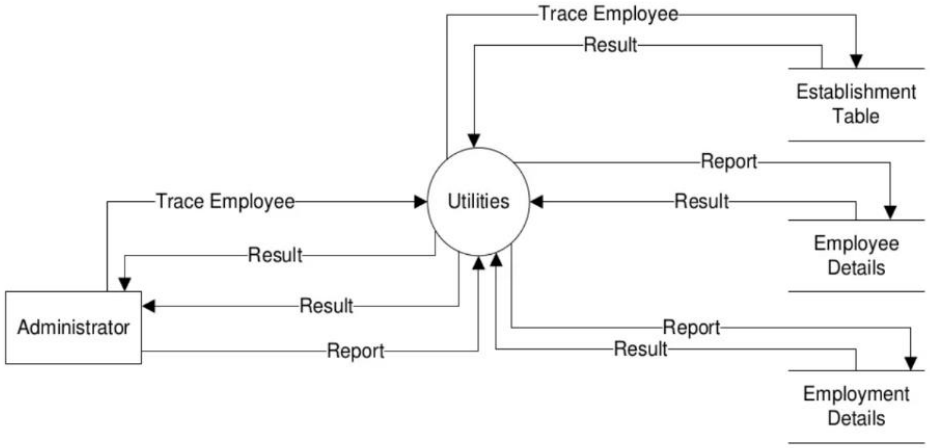
First Level Diagram :



Second Level Diagram (Adding Employee) :



Second Level Diagram (For tracing employee and reporting utility :



CHAPTER 2

2. AIM AND SCOPE OF EMS

In this world of growing technologies everything has been computerized. With large number of work opportunities the Human workforce has increased. Thus there is a need of a system which can handle the data of such a large number of Employees in an organization. This project simplifies the task of maintain records because of its user friendly nature.

2.1 PROCESS DATA

- Display- User with defined roles can display the content of the database. Being more specific, employee can only view his/her personal information. HOD can not only see his/her personal information but also employee's information who are under his/her department or school. Admin and HR can display their personal information and all employees' information.

- Edit- A user with employee role can edit his/her specific personal information. Dean or HOD can only edit employees' personal information that is under his/her coverage except user role type. Admin can edit all information related to all employees' including their user role type.

- Search- User with Dean/HOD role can search the content of database for the employees' who are under his/her coverage. HR and admin roles can search all the employees' information in the database. Search feature works on specific keywords showing employee's characteristics, peculiarities, skills, features, and etc. For example, HR wants to find employees' who are well trained in "Java Programming Language". He/she will write the specific keyword in the search bar and press the available search button. Afterwards, he/she will find a list of all the employees' who know "Java Programming". Update authentication- This feature can be used only by admin role type. Admin can update the role type of a specific user. For example, an employee got promotion and his role type will be changed from employee role id to HOD or Dean role. Admin will be able to update this authentication mechanism.

2.2 OBJECTIVES

In this world of growing technologies everything has been computerized. With large number of work opportunities the Human workforce has increased. Thus there is a need of a system which can handle the data of such a large number of Employees. This project simplifies the task of maintaining records because of its user friendly nature. The objective of this project is to provide a comprehensive approach towards the management of employee information. This will be done by designing and implementing an HR management system that will bring up a major paradigm shift in the way that employee information is handled.

The objectives of this system include:

- Design of a web based HR management system to fulfill requirements such as project management, leave management, report generation to assist in performance appraisal, ESS and employee trainings.
- Well-designed database to store employee information.
- A user friendly front-end for the user to interact with the system.

2.3 ADVANTAGES

This system is expected to be user friendly and will offer easy access to data as well as services such as online leave management, e-recruitment, and timely report generation, monitoring employee trainings, task management, project management and employee tracking. The employee is expected to have direct interaction with this system through a password protected user account therefore proposed system is web based to enable accessibility from any location as long as internet connectivity is available. This direct interaction with the system will enable employee self-service. Without an employee management system, it’s a tedious job for the human resource department to keep track of each and every employee and even harder for a project manager to assign tasks to the project team. The HR management system will be developed to provide information of employees and many other facilities at the click of a button.

DISADVANTAGES

Due to the constraint of resources and time, the size of the project could not be increased.

- Risk of Internal Competition. Under this system, employees compete with each other for job status, position and pay.
- Favouritism
- Expensive and Time-Consuming
- Manager's Dilemma
- Convoluted and Bureaucratic.

2.4 Hardware requirements

EMS should be able to work on a computer with the following minimum hardware specifications:
OS: Windows XP/Vista/7/8 and Linux
CPU: Pentium III (700MHz) and above
Memory: 128 MB and above
Capacity: 4GB of hard drive
Others: Network interface card, mouse, keyboard, and monitor

CHAPTER 3

3.1 DEVELOPMENT TOOLS

HTML

Hyper Text Markup Language (HTML) is a computer language devised to allow website creation. These websites can then be viewed by anyone else connected to the Internet. It is relatively easy to learn, with the basics being accessible to most people in one sitting; and quite powerful in what it allows you to create. Having the basic knowledge of HTML will could help make or develop m-commerce websites and will also prove to be handy especially for editing and modifying web pages. Furthermore, it has some low cost benefits because of its many free online tutorials and advice support which is vital for m-commerce development.

JAVASCRIPT

JavaScript is a scripting language that is browser based and was developed by Netscape to enable web masters/authors to add interactivity and enhances behavior of web pages. Some of the dynamic behavior that can be generated by JavaScript is validating form, performing specific actions e.g. after a mouse click, adding timestamps etc. JavaScript is an open language and anyone can use it. It also shares m any of the features and structures of the Java programming language, though it is not really related to Java. It was developed independently.

CSS

CSS is a style sheet language used to describe presentation and layout of HTML tags. CSS is used to enable separation of document content from document presentation. This refers to the separation of document presentation aspects such as colors, layouts and fonts from the actual document content. CSS helps us achieve layout design and control much easier.

JSON

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an object, record, struck, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

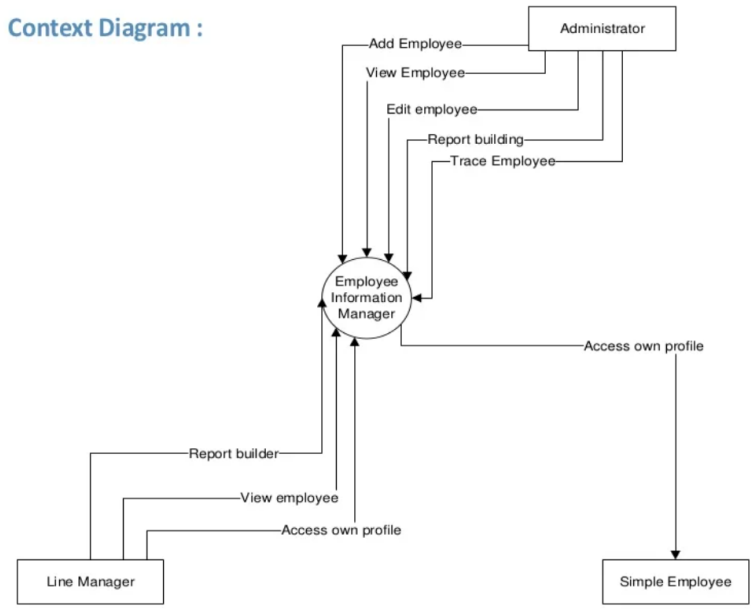
These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.

JQUERY

JQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

3.2 FLOW, SEQUENCE AND ACTIVITY DIAGRAM

FLOW DIAGRAM



Flow diagram

SEQUENCE DIAGRAM

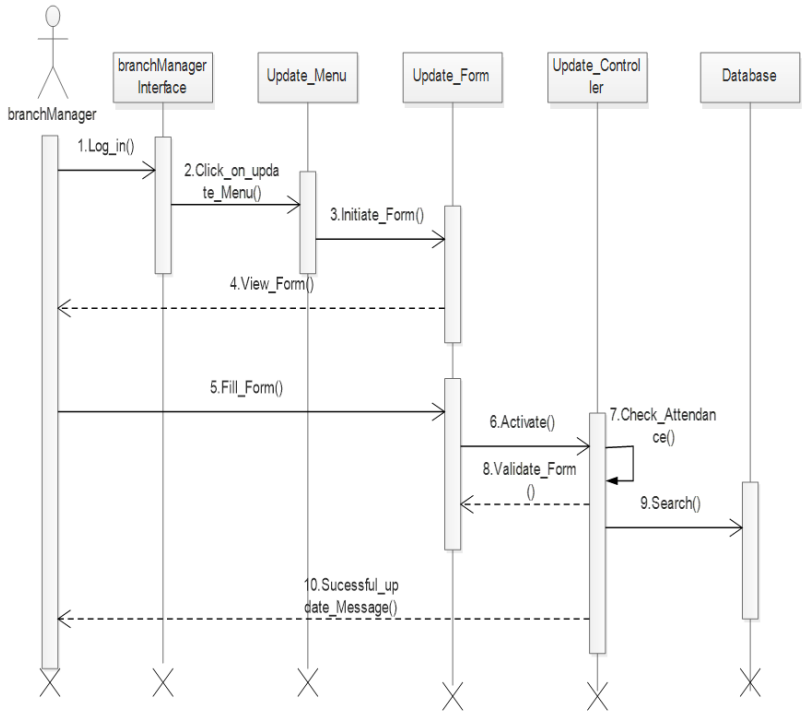
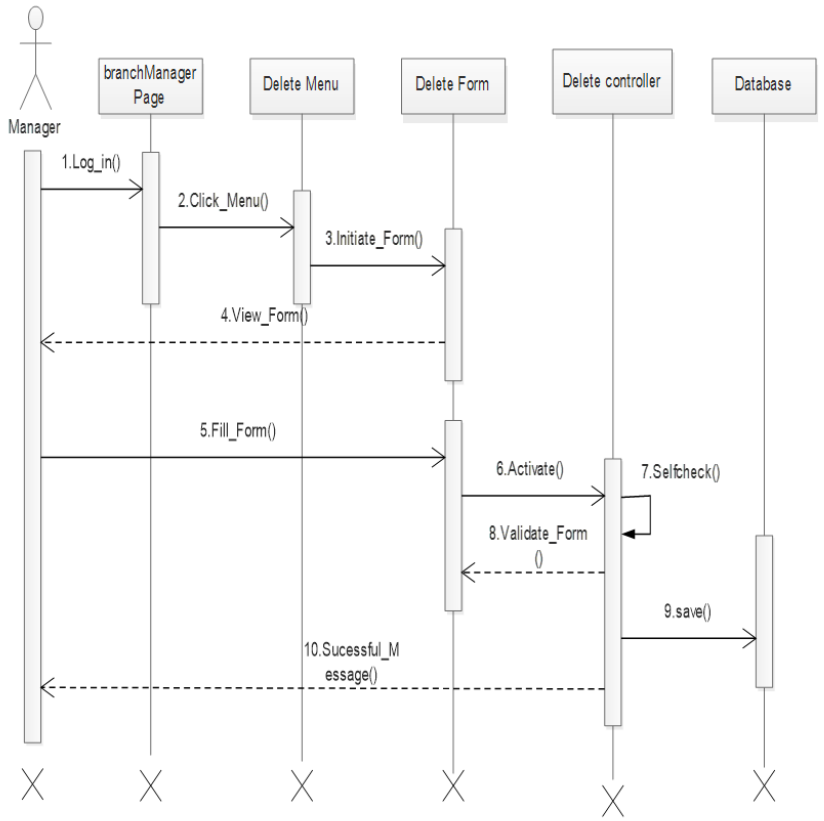
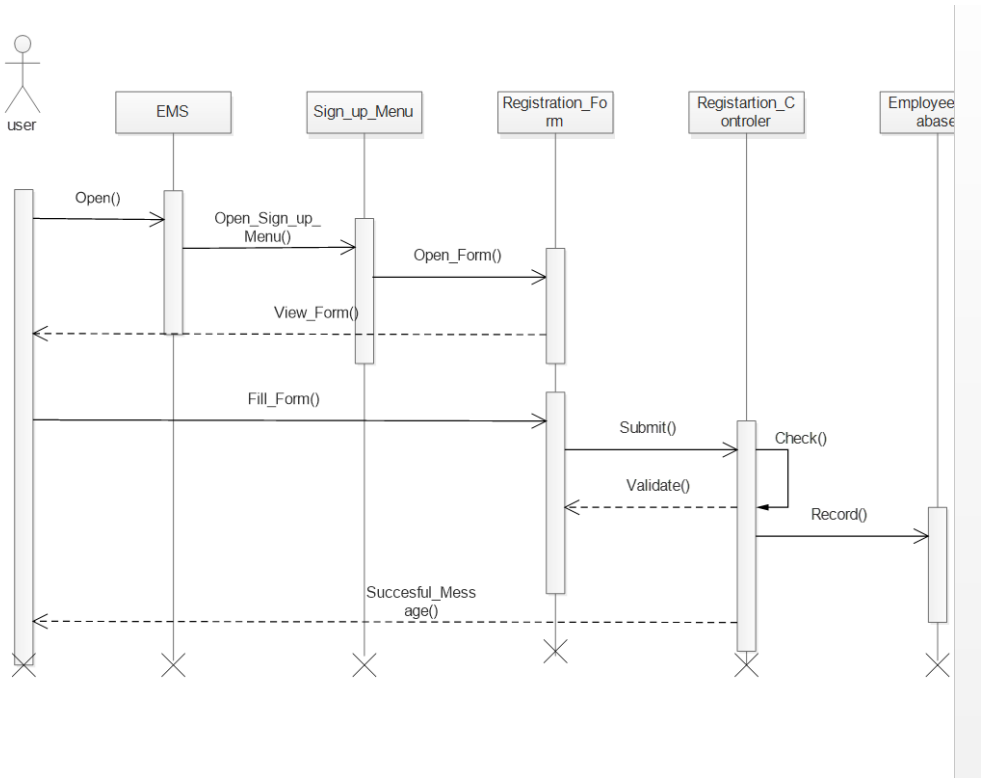


FIG 3.1 Update form



Delete form



Registration form

ACTIVITY DIAGRAM

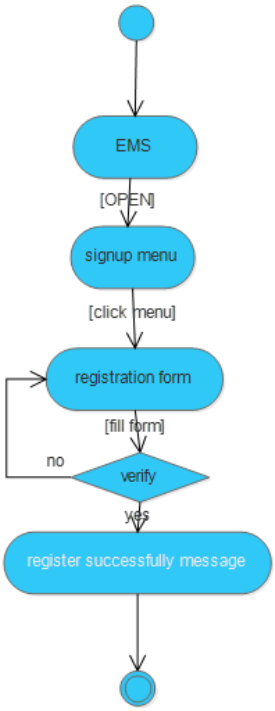
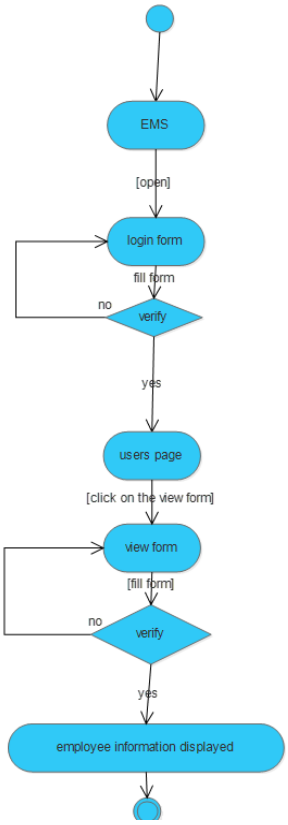
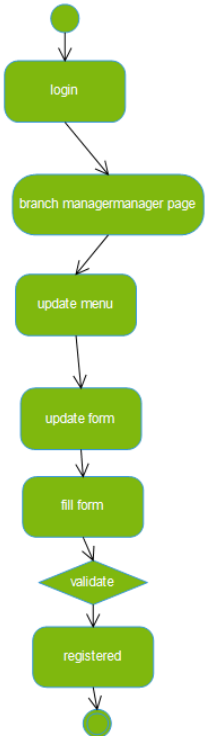


FIG 3.2 Registration



View record



State chart update record

USECASE DIAGRAM

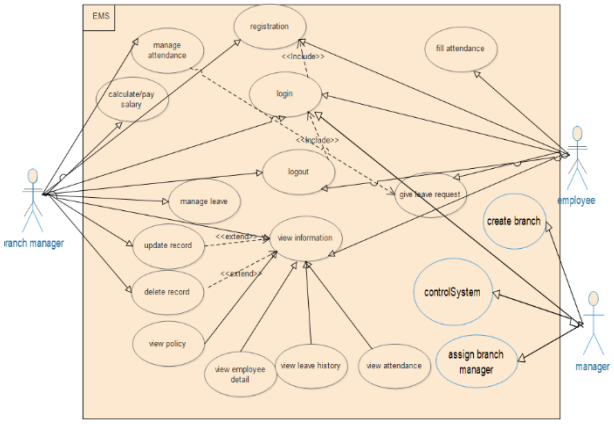


Fig 3.3 Usecase diagram

3.3 COMPONENTS

1) Use case name: registration

Goal: to register employee in the database

Flow event:

1. The employee or the manager open the system
2. The signup menu displayed on the system
3. The user clicks the menu and the registration form is displayed
4. Fill the complete information about his/her detail or the employee

Click register button

Exceptional flow:

If the employee /manager does not fill the correct information, the system notifies the user to enter the correct information

The system does not work when there is no connection**.**

Alternative flow: the employee can register contact physically with the manager.

2) Use case name: update record

Goal: to modify the records registered in the database.

Flow event:

1. The manager open and logged in to the system.
2. The update menu displayed on the system.
3. The user clicks the menu and the form is displayed.
4. The manager enters the information on the update form.
5. Click update button.
6. The detail is updated successfully message is displayed.

Exceptional flow:

If the manager does not fill the correct information in the form, please error displayed.

If the manager does not fill the form the system displays, please fill the form message is displayed

The system does not work when there is no connection. **Alternative flow:** none.

3.4 WORKING OF EMPLOYEE MANAGEMENT SYSTEMS

Used vscode to create the web application

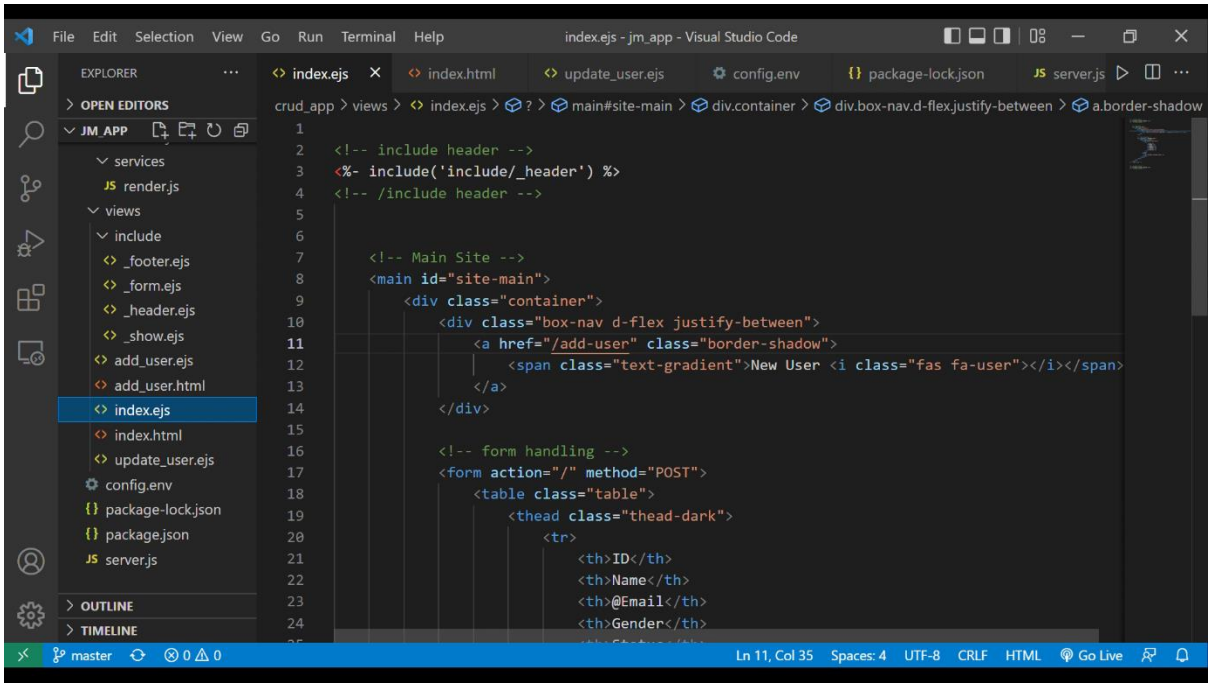
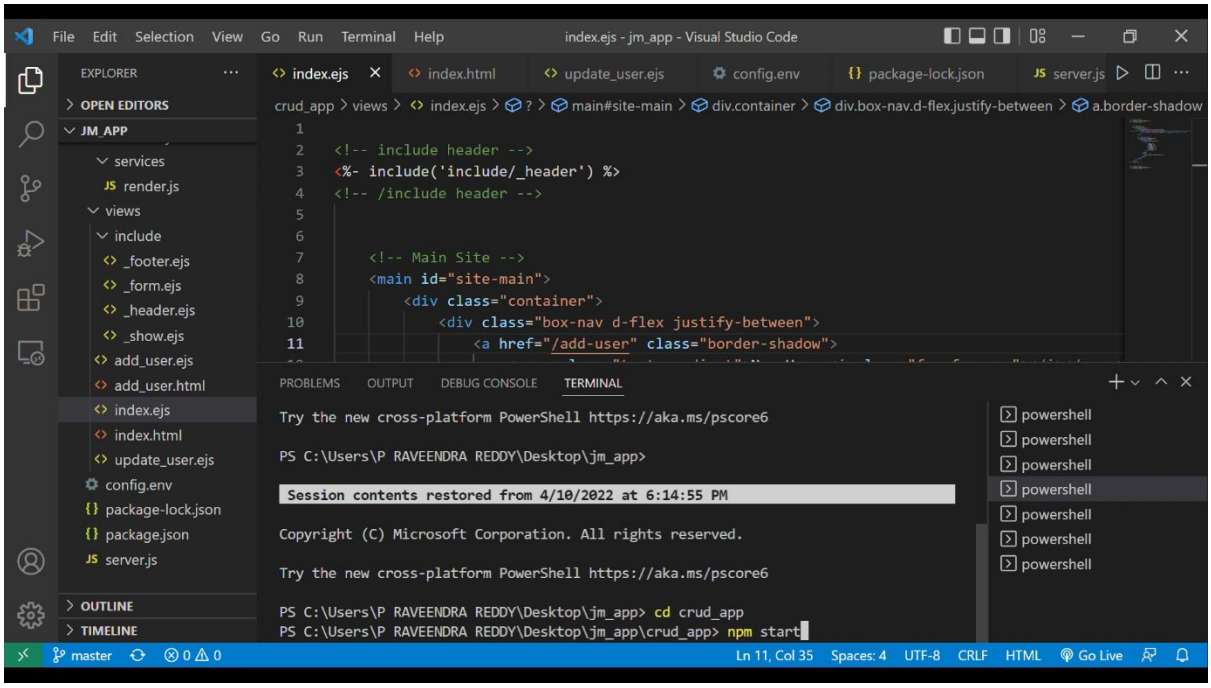
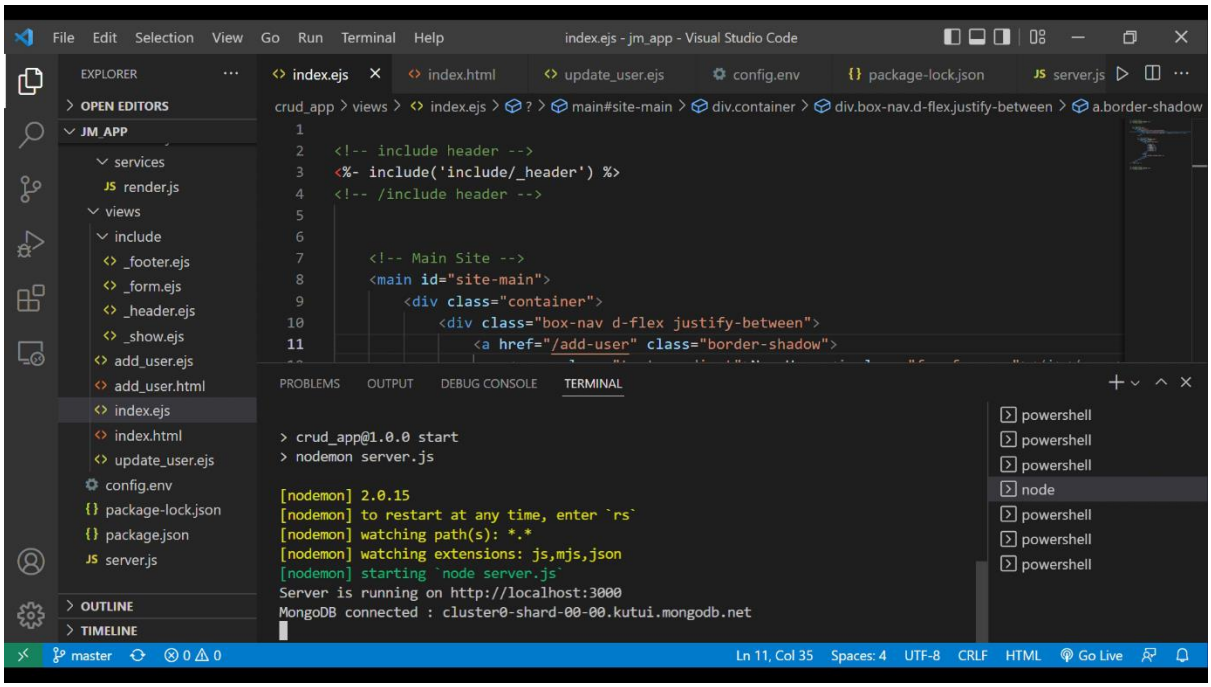


Fig 3.4 WORKING OF EMS

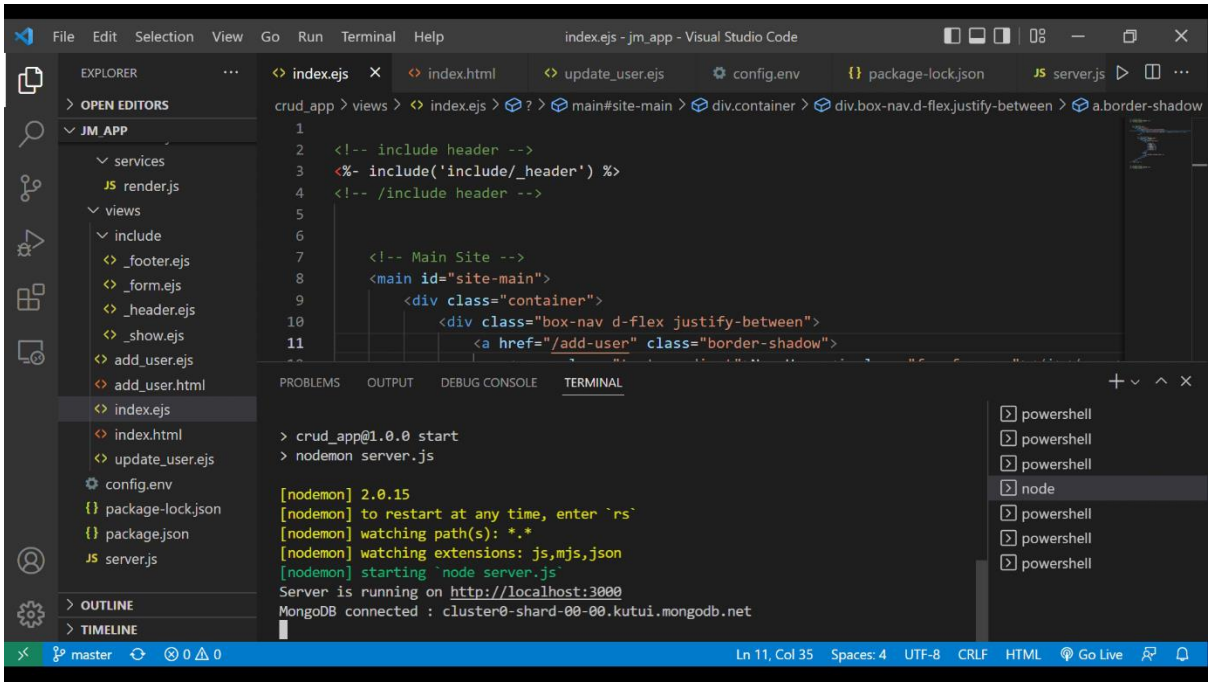
Open the terminal and type npm start to run the program



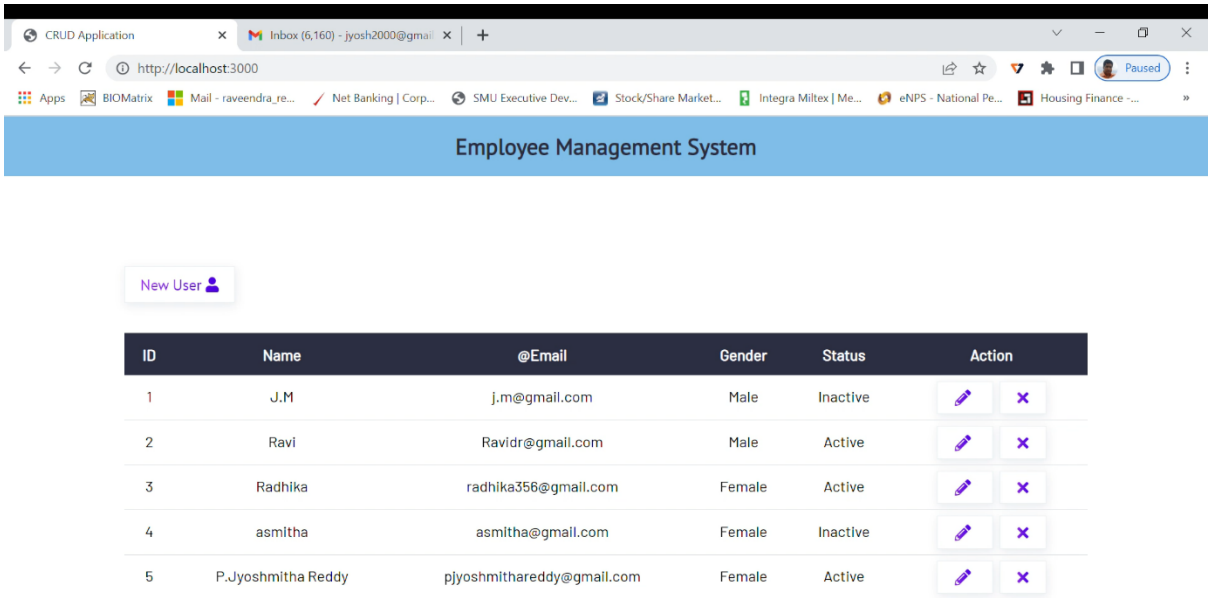
Now the server is running on localhost and is connected to mongodb



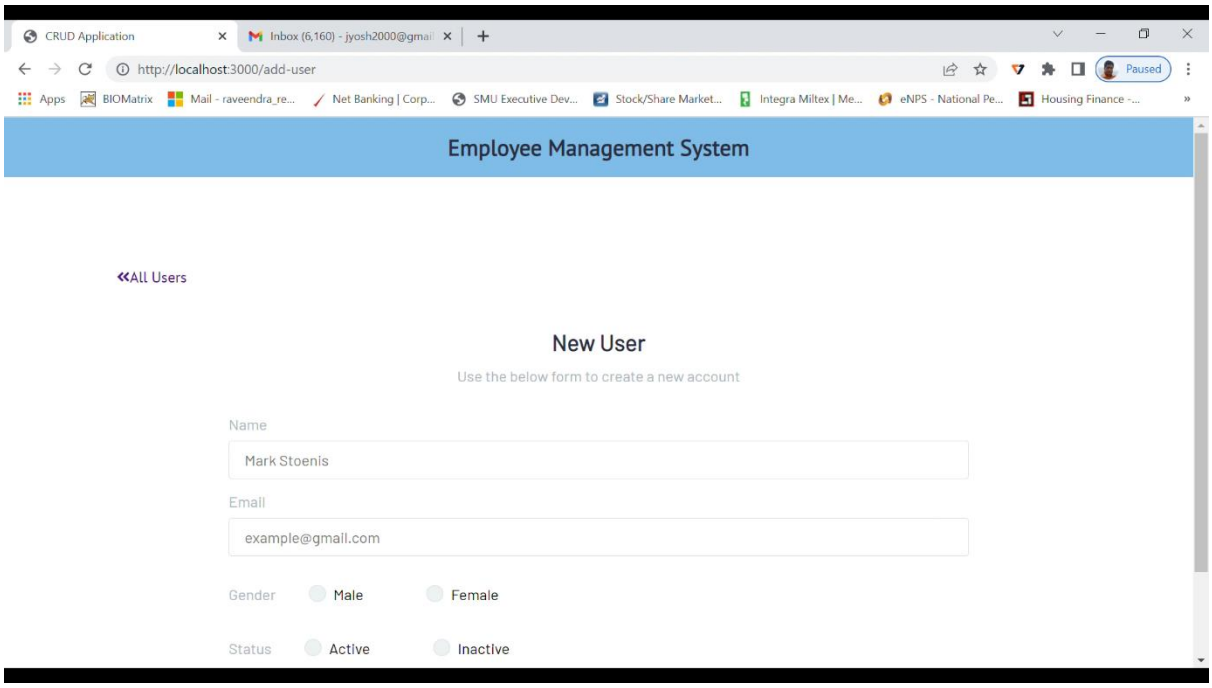
Click on localhost: 3000



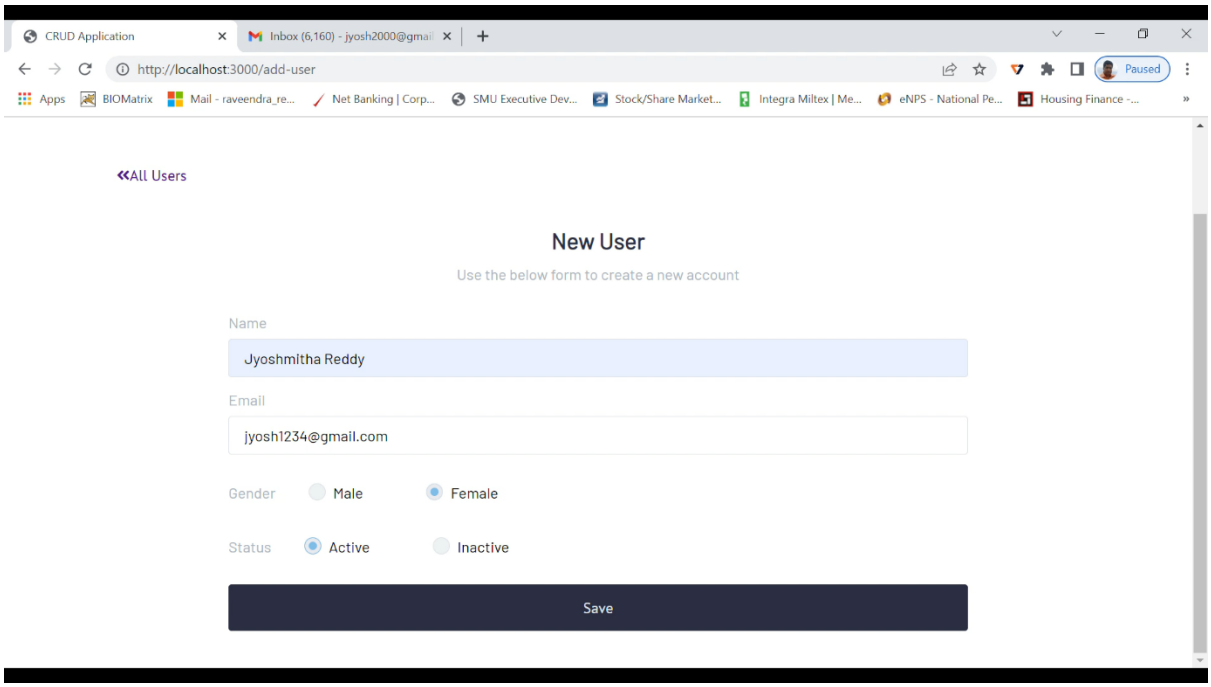
Now the app is running on the localhost:3000



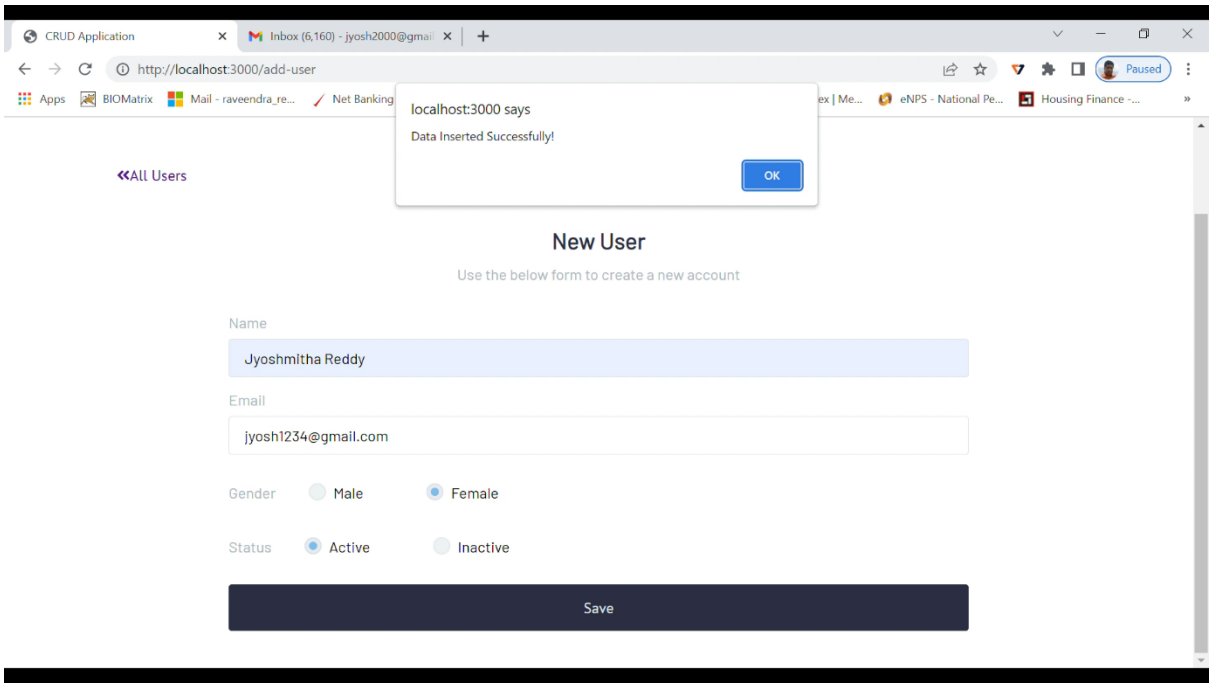
You can click on add users to add an employee



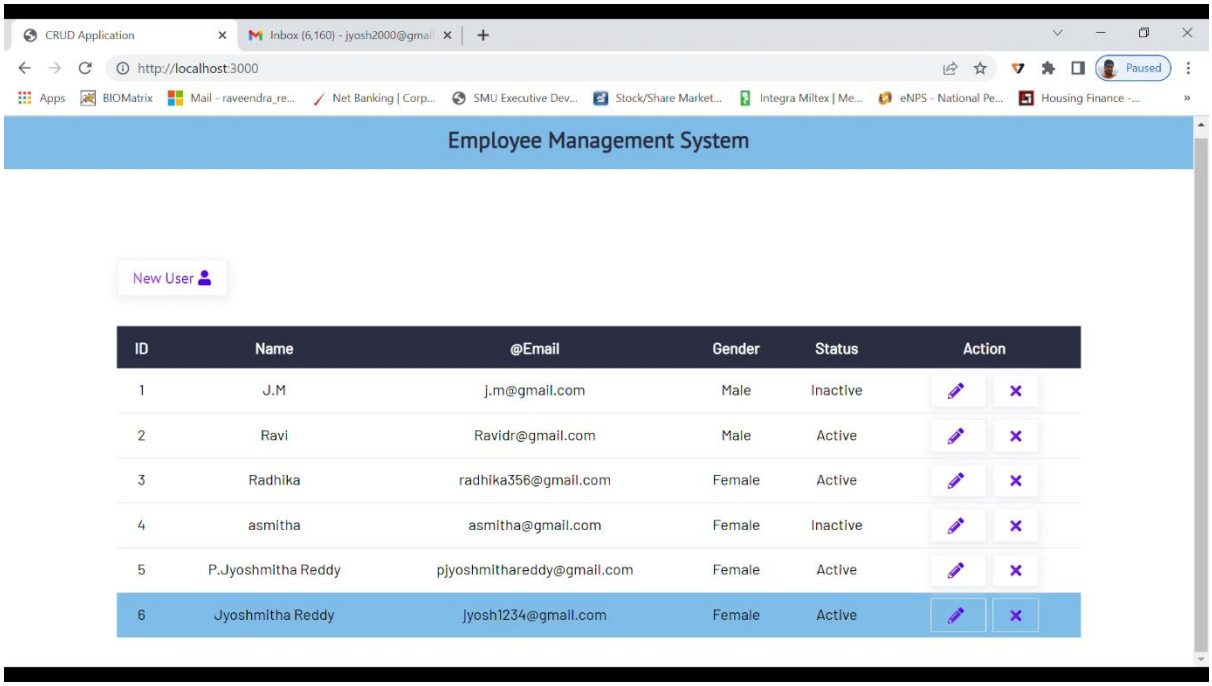
After filling the details click on save



You will get a message saying data inserted successfully

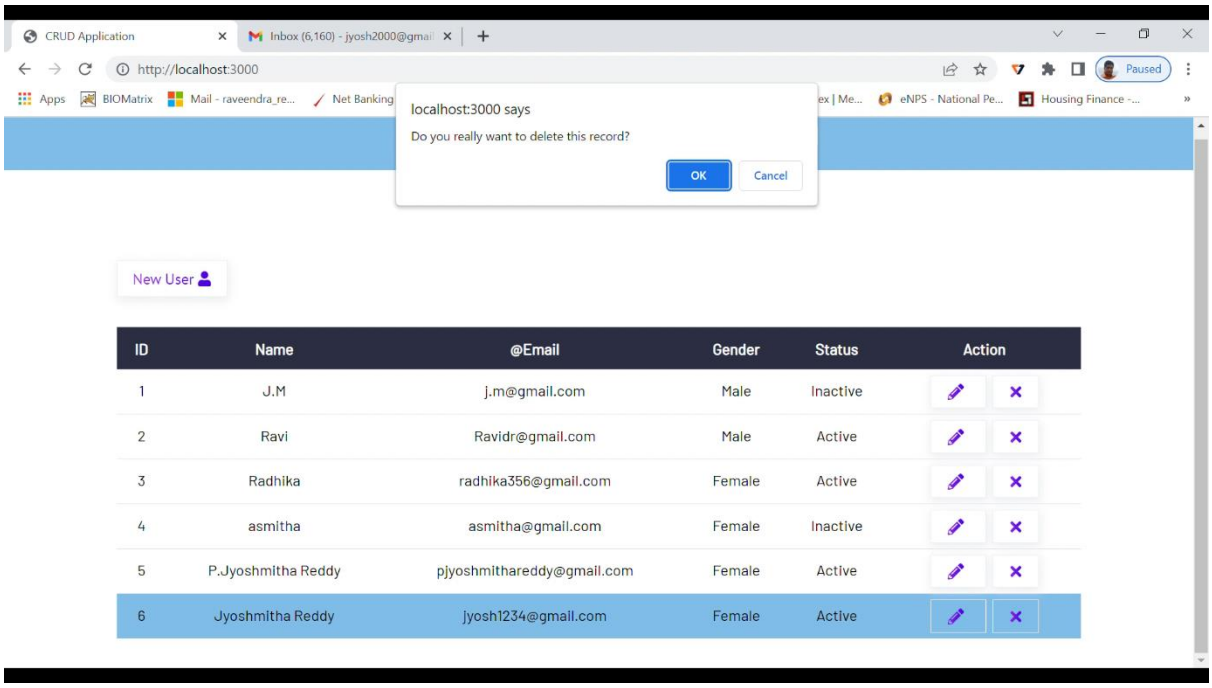


You can now view the new user

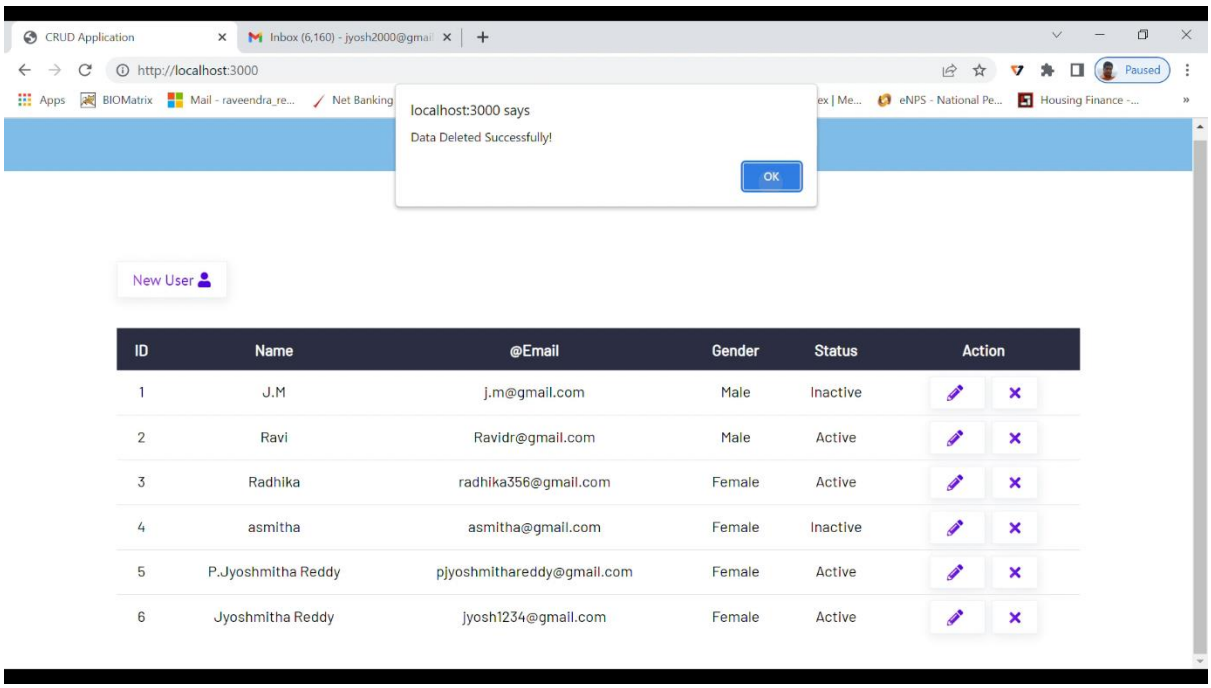


ID	Name	@Email	Gender	Status	Action
1	J.M	j.m@gmail.com	Male	Inactive	
2	Ravi	Ravidr@gmail.com	Male	Active	
3	Radhika	radhika356@gmail.com	Female	Active	
4	asmitha	asmitha@gmail.com	Female	Inactive	
5	P.Jyoshmitha Reddy	pjyoshmithareddy@gmail.com	Female	Active	
6	Jyoshmitha Reddy	jyosh1234@gmail.com	Female	Active	

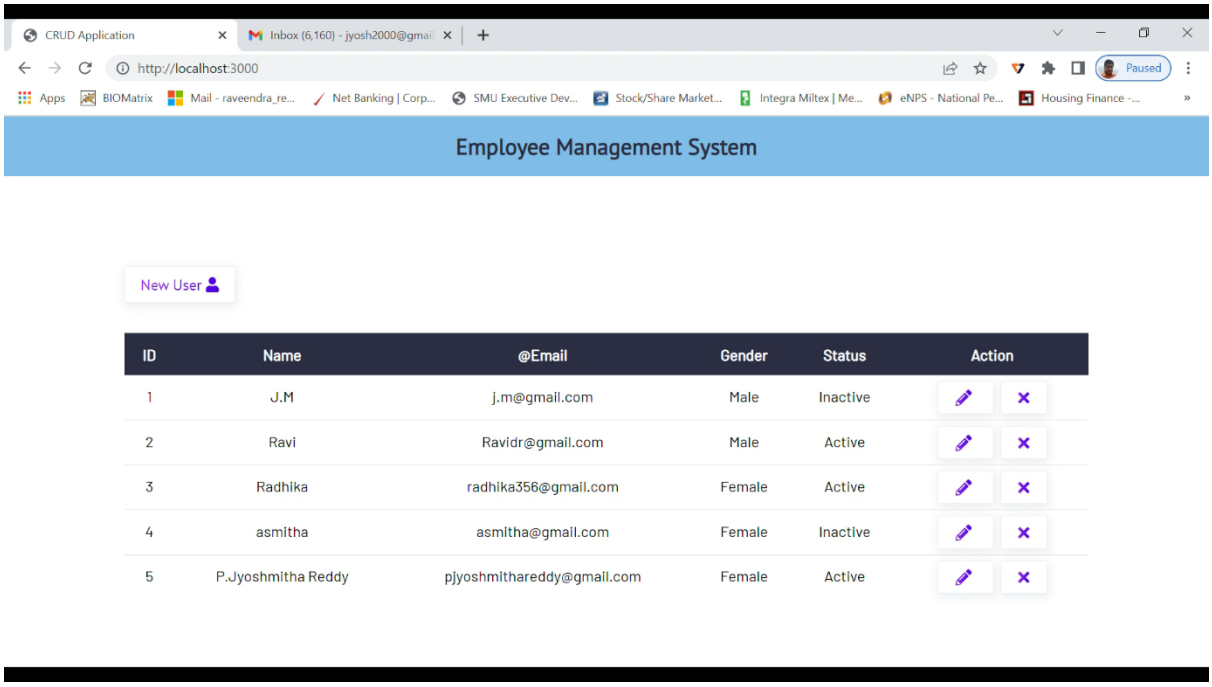
You can also delete an user



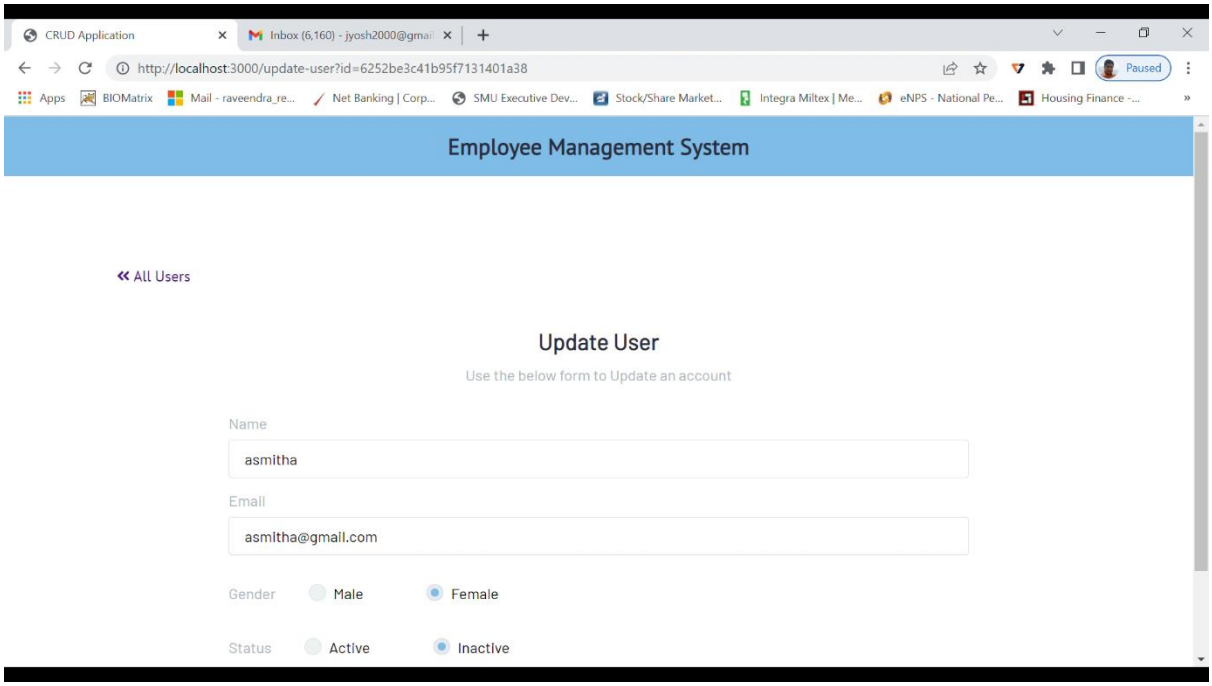
ID	Name	@Email	Gender	Status	Action
1	J.M	j.m@gmail.com	Male	Inactive	
2	Ravi	Ravidr@gmail.com	Male	Active	
3	Radhika	radhika356@gmail.com	Female	Active	
4	asmitha	asmitha@gmail.com	Female	Inactive	
5	P.Jyoshmitha Reddy	pjyoshmithareddy@gmail.com	Female	Active	
6	Jyoshmitha Reddy	jyosh1234@gmail.com	Female	Active	



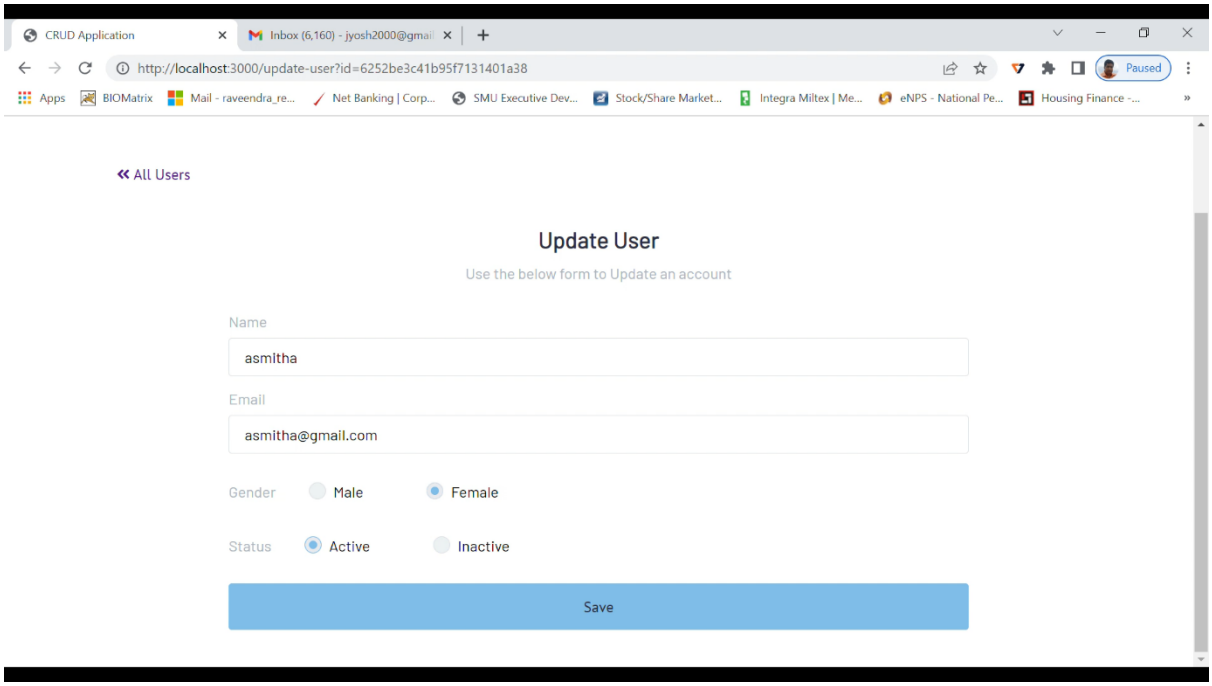
You can update the users info by clicking the update icon

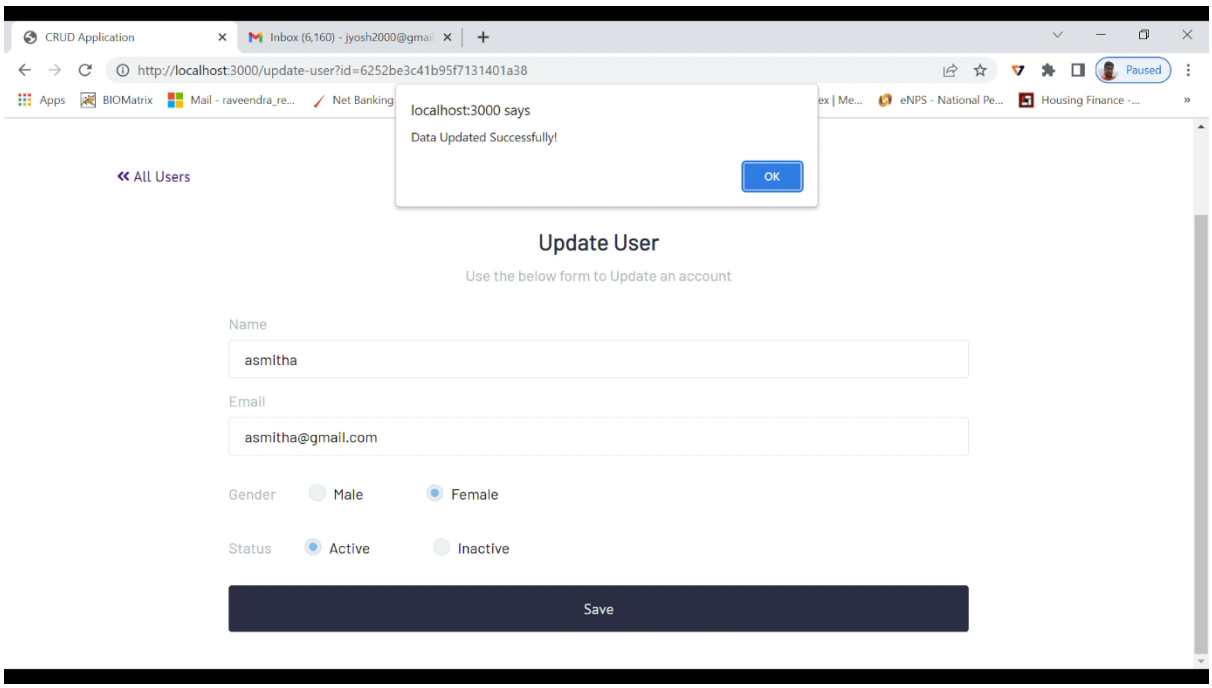


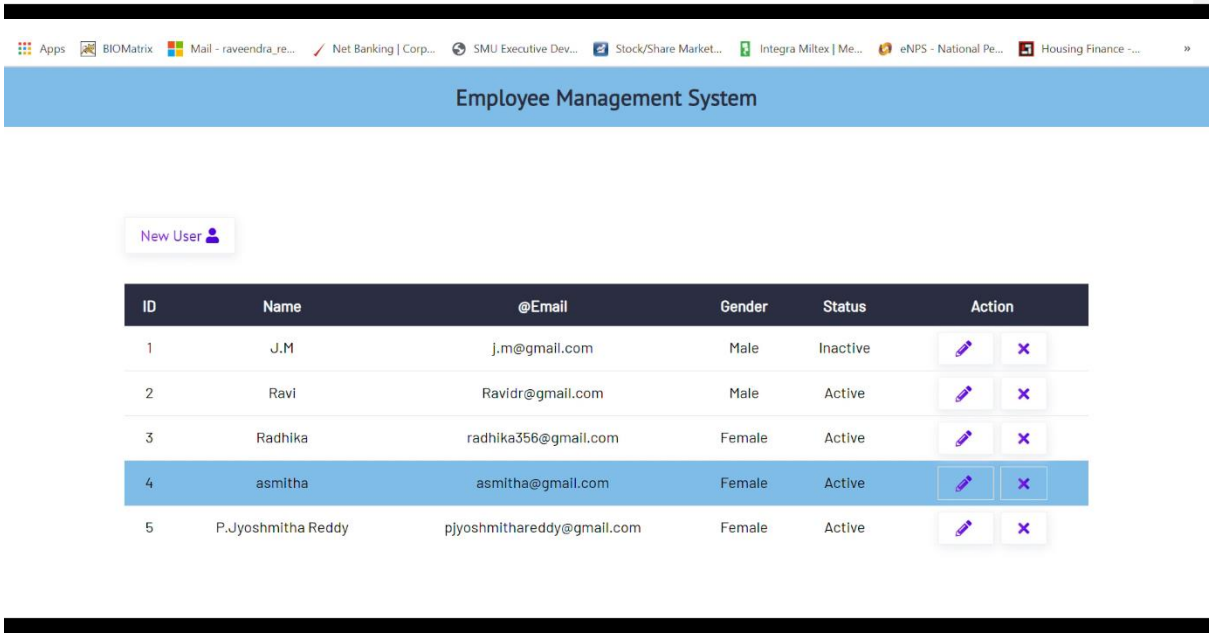
Change the details

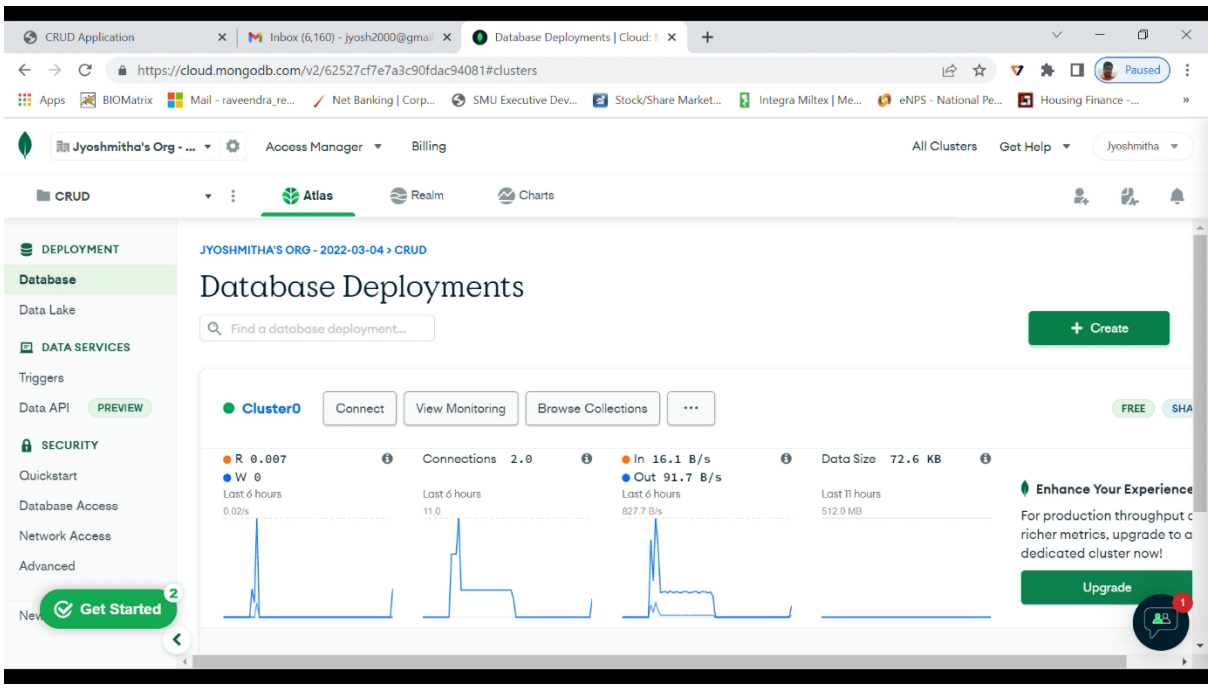


And save

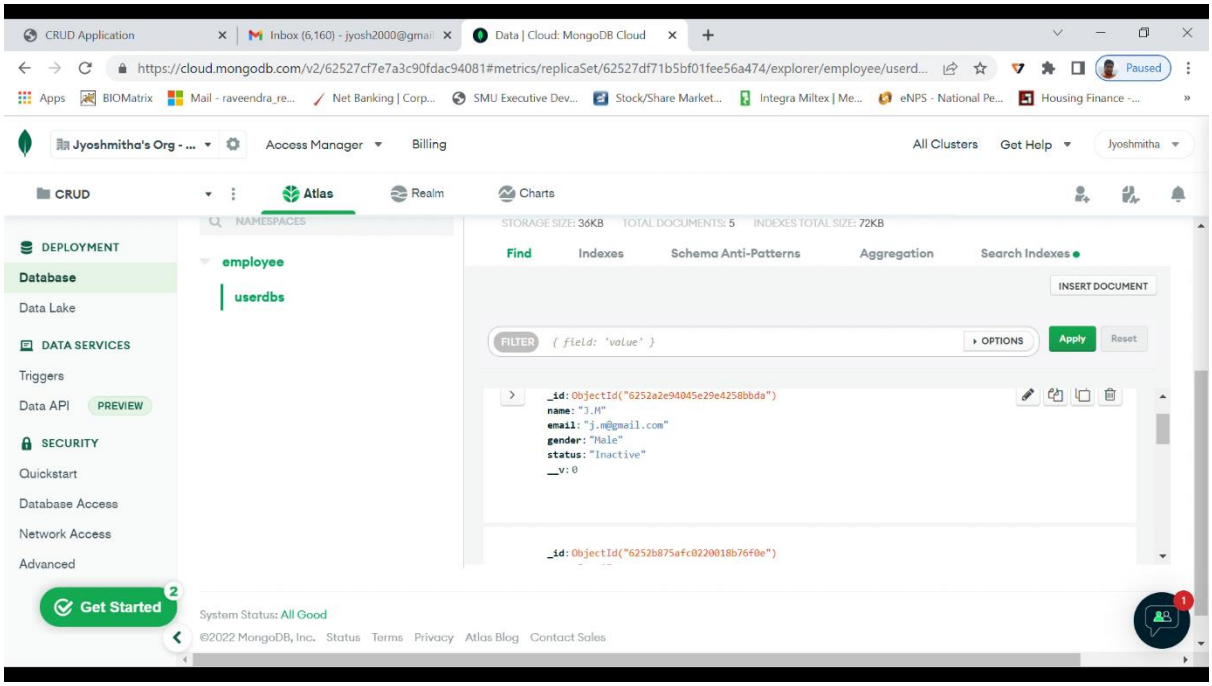


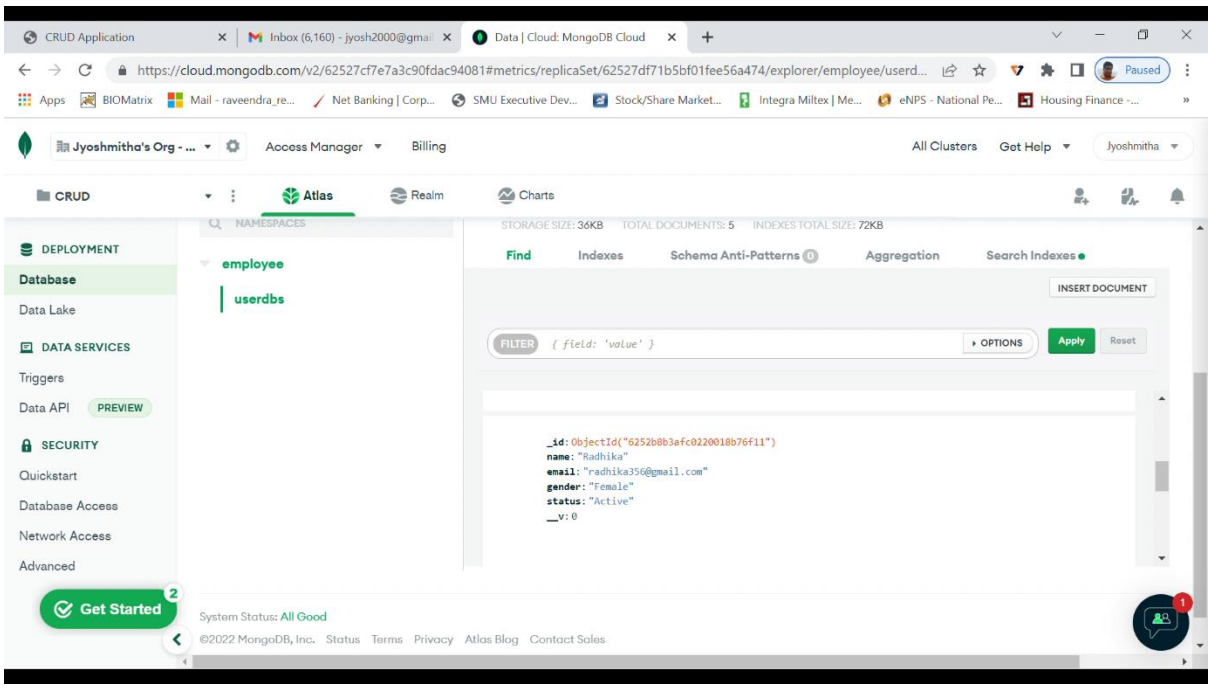

CRUD Application
Inbox (6,160) - jyosh2000@gmail
http://localhost:3000/update-user?id=6252be3c41b95f7131401a38
localhost:3000 says
Data Updated Successfully!
OK
All Users
Update User
Use the below form to Update an account
Name
asmitha
Email
asmitha@gmail.com
Gender
Male
Female
Status
Active
Inactive
Save

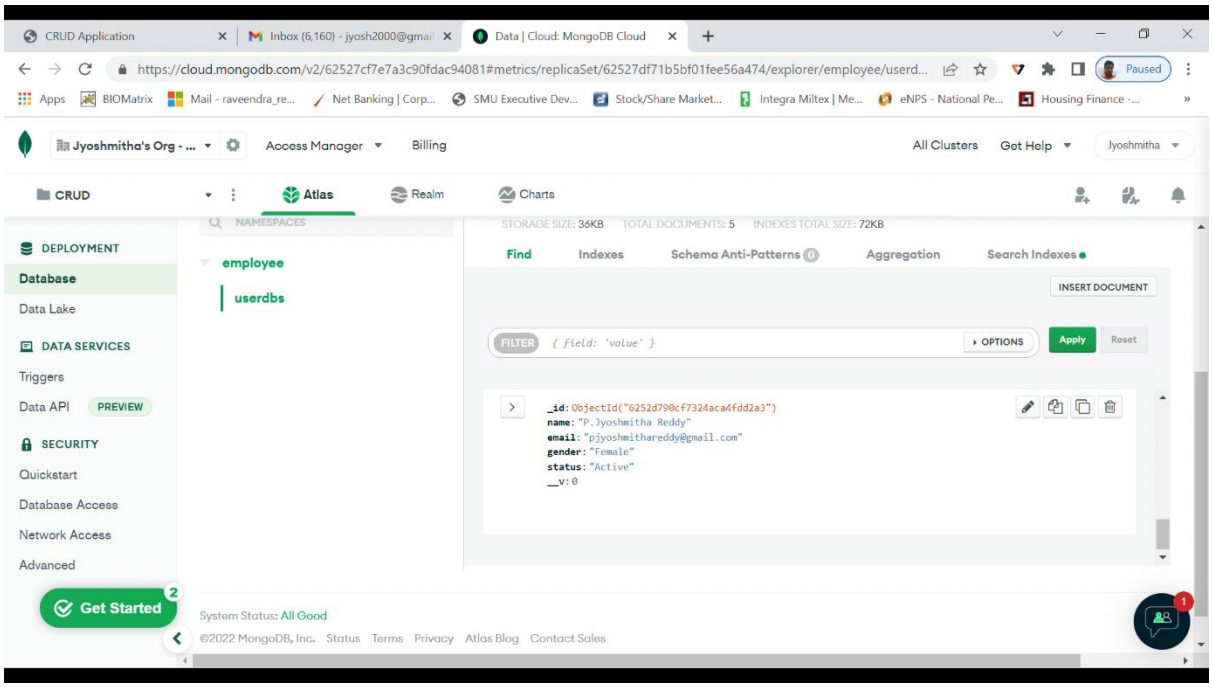

Employee Management System
New User
ID
Name
@Email
Gender
Status
Action
1 J.M j.m@gmail.com Male Inactive
2 Ravi Ravidr@gmail.com Male Active
3 Radhika radhika356@gmail.com Female Active
4 asmitha asmitha@gmail.com Female Active
5 P.Jyoshmitha Reddy pjyoshmithareddy@gmail.com Female Active



You can view all your data in the mongo dB database




CRUD Application
Inbox (6,160) - jyosh2000@gmail
Data | Cloud: MongoDB Cloud
https://cloud.mongodb.com/v2/62527cf7e7a3c90fdac94081#metrics/replicaSet/62527df71b5bf01fee56a474/explorer/employee/userd...
Jyoshmitha's Org - ...
Access Manager
Billing
All Clusters
Get Help
Jyoshmitha
CRUD
Atlas
Realm
Charts
DEPLOYMENT
Database
Data Lake
DATA SERVICES
Triggers
Data API
PREVIEW
SECURITY
Quickstart
Database Access
Network Access
Advanced
Get Started
employee
userdbs
STORAGE SIZE: 36KB
TOTAL DOCUMENTS: 5
INDEXES TOTAL SIZE: 72KB
Find
Indexes
Schema Anti-Patterns
Aggregation
Search Indexes
INSERT DOCUMENT
FILTER
{ field: 'value' }
OPTIONS
Apply
Reset
_id: ObjectId("6252b8b3afc0220018b76f11")
name: "Radhika"
email: "radhika356@gmail.com"
gender: "Female"
status: "Active"
__v: 0
System Status: All Good
©2022 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales


CRUD Application
Inbox (6,160) - jyosh2000@gmail
Data | Cloud: MongoDB Cloud
https://cloud.mongodb.com/v2/62527cf7e7a3c90fdac94081#metrics/replicaSet/62527df71b5bf01fee56a474/explorer/employee/userd...
Jyoshmitha's Org - ...
Access Manager
Billing
All Clusters
Get Help
Jyoshmitha
CRUD
Atlas
Realm
Charts
DEPLOYMENT
Database
Data Lake
DATA SERVICES
Triggers
Data API
PREVIEW
SECURITY
Quickstart
Database Access
Network Access
Advanced
Get Started
employee
userdbs
STORAGE SIZE: 36KB
TOTAL DOCUMENTS: 5
INDEXES TOTAL SIZE: 72KB
Find
Indexes
Schema Anti-Patterns
Aggregation
Search Indexes
INSERT DOCUMENT
FILTER
{ field: 'value' }
OPTIONS
Apply
Reset
_id: ObjectId("6252d790cf7324aca4fdd2a3")
name: "P.Jyoshmitha Reddy"
email: "pjyoshmithareddy@gmail.com"
gender: "Female"
status: "Active"
__v: 0
System Status: All Good
©2022 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

CHAPTER 4

4. RESULTS, DISCUSSION AND PERFORMANCE ANALYSIS

4.1 TESTING

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a probability of finding an as yet undiscovered error. A successful test is one that uncovers an as yet undiscovered error. Our Objective is to design test processes that systematically uncover different classes of errors and do so with minimum amount of time and effort. Generally, the main objective of testing is finding faults.

A description of the scope of the software testing is developed. All the features to be tested are noted as follows. The basic principles that guides software testing are,

- ✓ All test cases should be traceable top employee requirements. The most severe defects from the employee's point of view are those that cause the program to fail to meet its requirements.
- ✓ Test case should be planned long before testing begins. Testing plan can begin as soon as the requirement model is complete. Detailed definition of the test cases can begin as soon as the design is solidified. Therefore, the entire test can be planned before any code has been generated.
- ✓ Testing should begin "in the small" and progress towards "in the large". The first test planned and executed generally focus on the individual modules. As testing progresses testing shifts focus in an attempt to find errors in integrating clusters of modules and ultimately in the entire system
- ✓ Generally, before doing any activity of the system we try to test whether the user is login or not,
- ✓ We also test employee fill attendance once a day or not, the employee salary is payed once in a month or not, the employee who takes the salary is the member of the university or not.

4.2 PERFORMANCE ANALYSIS

The project team recommend some ideas considering the project. This project is developed based on the current problems of employee management system of ASTU, so we think that this project is the best solution for the problems raised based on the manual system of the office. And help the office to save time, money and man power whenever the project is small scale. Finally, the project teams conclude that the department should give a special consideration for the computer lab, because during the development of the project the group members faced some problems based on the lab.

4.3 PROJECT TIME LINE

The following table shows the expected flow of work for the accomplishment of the required result.

No.	Description	Duration	Status
1.	Research	1 day	Done
2.	Implementing code	1 day	Done
3.	Testing, debugging and error removal	1 day	Done
4.	Preparation of final report	5 days	Done

Table.4.1: Project Time Line

4.4 RESULT

The software product produced was fairly good, it achieved most of the user requirements, the user interface is good and is very easy to navigate, and even novice users can find their way around the web application easily. The client side validation is excellent.

CHAPTER 5

5. SUMMARY AND CONCLUSIONS

5.1 SUMMARY

The employee management system project is proposed to effectively understand the work, the type of person who is fit for the job, and the organization. It empowers the employee to accomplish the job and manages employees very well.

This project assisted me to gain a practical experience and apply the knowledge assimilated from the previous courses undertook. Putting the knowledge gained earlier and applying different techniques from past courses was interesting and certain concepts, tools and techniques only made sense after seeing their application in a real world scenario. It was extremely challenging at times but it has been a great and worthwhile learning experience. There is not at all any doubt that the employee management system would be an asset to any company, small or large.

5.2 CONCLUSION

The objective of this project was to build a web based program for EMS in order to manage employees safely and securely. The system developed is able to meet all the basic requirements. It will provide the facility to the end-user and staffs members of the office. The EMS will be also benefited by the proposed system, as it will automate the whole procedure, which will reduce the workload. The security of the system is also one of the prime concerns.

There is always a room for improvement in any software, however efficient the system may be. The important thing is that the system should be flexible enough for future modifications and to maintain. Every effort has been made to cover all user requirements and make it user friendly.

REFERENCES

- www.google.com
- www.wikipedia.com
- www.youtube.com

APPENDIX

Source Code:

Index.ejs

```
<!-- include header -->
<%- include('include/_header') %>
<!-- /include header -->


  <!-- Main Site -->
  <main id="site-main">
    <div class="container">
      <div class="box-nav d-flex justify-between">
        <a href="/add-user" class="border-shadow">
          <span class="text-gradient">New User <i class="fas fa-
user"></i></span>
        </a>
      </div>

      <!-- form handling -->
      <form action="/" method="POST">
        <table class="table">
          <thead class="thead-dark">
            <tr>
```

```
                    <th>ID</th>
                    <th>Name</th>
                    <th>@Email</th>
                    <th>Gender</th>
                    <th>Status</th>
                    <th>Action</th>
                  </tr>
                </thead>
                <tbody>
                  <%- include('include/_show') %>
                </tbody>
              </table>
            </form>
          </div>
        </main>
  <!-- /Main Site -->



  <!-- include footer -->
  <%- include('include/_footer') %>
  <!-- /include footer -->
```

Add_user.ejs

```
        <!-- include header -->
        <%- include('include/_header') %>
        <!-- /include header -->

        <!-- Main Site -->
        <main id="site-main">
          <div class="container">
            <div class="box-nav d-flex justify-between">
              <div class="filter">
                <a href="/"><i class="fas fa-angle-double-left"></i>All
```

```
Users</a>
            </div>
        </div>
        <div class="form-title text-center">
            <h2 class="text-dark">New User</h2>
            <span class="text-light">Use the below form to create a new
account</span>
        </div>

        <!-- add user form -->
        <%- include('include/_form') %>

    </div>
  </main>
  <!-- /Main Site -->



<!-- include footer -->
<%- include('include/_footer') %>
<!-- /include footer -->
```

Update_user.ejs

```
<!-- include header -->
<%- include('include/_header') %>
<!-- /include header -->

  <!-- Main Site -->
  <main id="site-main">
    <div class="container">
        <div class="box-nav d-flex justify-between">
          <div class="filter">
              <a href="/"><i class="fas fa-angle-double-left"></i> All
Users</a>
```

```
        </div>
      </div>
      <div class="form-title text-center">
        <h2 class="text-dark">Update User</h2>
        <span class="text-light">Use the below form to Update an account</span>
      </div>

      <!-- add user form -->
       <!-- form handling -->
 <form method="POST" id="update_user">
   <div class="new_user">
      <div class="form-group">
        <label for="name" class="text-light">Name</label>
        <input type="hidden" name="id" value="<%= user._id %>">
        <input type="text" name="name" value="<%= user.name %>" placeholder="Mark Stoenis">
      </div>
      <div class="form-group">
        <label for="Email" class="text-light">Email</label>
        <input type="text" name="email" value="<%= user.email%>" placeholder="example@gmail.com">
      </div>
      <div class="form-group">
        <label for="gender" class="text-light">Gender</label>
        <div class="radio inline">
          <input type="radio" id="radio-2" name="gender" value="Male" <%= user.gender == 'Male' ? 'checked' : '' %>>
          <label for="radio-2" class="radio-label">Male</label>
        </div>
        <div class="radio inline">
          <input type="radio" id="radio-3" name="gender" value="Female"  <%= user.gender == 'Female' ? 'checked' : '' %> >
          <label for="radio-3" class="radio-label">Female</label>
        </div>
```

```
            </div>

            <div class="form-group">
                <label for="gender" class="text-light">Status</label>
                <div class="radio inline">
                    <input type="radio" id="radio-4" name="status" value="Active"
        <%= user.status == 'Active' ? 'checked' : '' %> >
                    <label for="radio-4" class="radio-label">Active</label>
                </div>
                <div class="radio inline">
                    <input type="radio" id="radio-5" name="status"
        value="Inactive" <%= user.status == 'Inactive' ? 'checked' : '' %> >
                    <label for="radio-5" class="radio-label">Inactive</label>
                </div>
            </div>

            <div class="form-group">
                <button type="submit" class="btn text-dark
        update">Save</button>
            </div>

        </div>
        </form>

        </div>
        </main>
        <!-- /Main Site -->

        <!-- include footer -->
        <%- include('include/_footer') %>
<!-- /include footer -->
```

Server.js

```
const express = require('express');
const dotenv = require('dotenv');
const morgan = require('morgan');
const bodyparser = require("body-parser");
const path = require('path');

const connectDB = require('./server/database/connection');

const app = express();

dotenv.config( { path : 'config.env'} )
const PORT = process.env.PORT || 8080

app.use(morgan('tiny'));

connectDB();

app.use(bodyparser.urlencoded({ extended : true}))

app.set("view engine", "ejs")

app.use('/css', express.static(path.resolve(__dirname, "assets/css")))
app.use('/img', express.static(path.resolve(__dirname, "assets/img")))
app.use('/js', express.static(path.resolve(__dirname, "assets/js")))

app.use('/', require('./server/routes/router'))

app.listen(PORT, ()=> { console.log(`Server is running on http://localhost:${PORT}`)});
```