DOCUMENT SUBMISSION SYSTEM

A
MAJOR PROJECT-I REPORT

Submitted by

Himanshu Pathak	Varad Khokle	Anant Agarwal	Shashank Mishra
0902CS191016	0902CS191060	0902CS191004	0902CS191048

B.Tech. (COMPUTER SCIENCE AND ENGINEERING)
(Batch 2019-23)

Under the Guidance of
Dr.Jagdish Makhijani
Assistant Professor & HOD (Computer Science &Engineering)

DECEMBER-2022

DEPARTMENT OF COMPUTER SCIENCE &ENGINEERING
RUSTAMJI INSTITUTE OF TECHNOLOGY, BSF ACADEMY, TEKANPUR

Github repository url:
https://github.com/Himanshupathak13/PPS/tree/himanshu/testing

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY
RUSTAMJI INSTITUTE OF TECHNOLOGY, BSF ACADEMY, TEKANPUR

CERTIFICATE

This is to Certified that this project report **DOCUMENT SUBMISSION SYSTEM** is submitted by Himanshu Pathak, Anant Agarwal, Varad Khokle, Shashank Mishra who carried out the project work under my supervision. I approve this project for submission of the Bachelor of Engineering in the Department of Computer Science & Information Technology, Rustamji Institute of Technology, BSF Academy, Tekanpur.

Dr. Jagdish Makhijani
(Project Supervisor)

Dr. Jagdish Makhijani
(Major Project-I Incharge)

Head of Department

Dr.Rashmi Shah
Dean Academics

ACKNOWLEDGEMENT

It gives me immense pleasure to express my deepest sense of gratitude and sincere thanks to my highly respected and esteemed guide Dr.Jagdish Makhijani, Assistant Professor & HOD (Computer Science & Engineering), Rustamji Institute of Technology, BSF Academy, Tekanpur, for their valuable guidance, encouragement and help for completing this work. Their useful suggestions for this whole work and co-operative behavior are sincerely acknowledged.

I would like to express my sincere thank to Shri Ajeeth Kumar P., Chief Administrator, RJIT, Tekanpur for giving me this opportunity to undertake this project. I would also like to thank Dr.Raghunath Sambhaji Holambe,Principal for whole hearted support. I would also like to thank Dr.Rashmi Shah, Dean Academics for whole hearted support.

I also wish to express my gratitude to Dr.Jagdish Makhijani, Assistant Professor &HOD (Computer Science &Engineering) for his kind hearted support. I am also grateful to my teachers Prof.Yograj Sharma, Prof.Kirtiraj Bhatele, Prof.Suhel khan for their constant support and guidance.

I also wish to express my indebtedness to my parents as well as my family members whose blessings and support always helped me to face the challenges ahead.

At the end I would like to express my sincere thanks to all my friends and others who helped me directly or indirectly during this project work.

Place: RJIT, Tekanpur — Himanshu Pathak

Date: 6 Dec. 2022 — 0902CS191016

TABLE OF CONTENTS

1. INTRODUCTION

1.1 INTRODUCTION :- The Project Document Submission System mainly deals with the uploading of project files by the registered user on the website. Under this system, a website is designed and developed to allow users to send and receive document files online. It also has an admin module to manage the documents uploaded on the system. The administrator of the system can approve or reject the uploaded documents.

Two major components of this website are:-

- User Onboarding (Registration and Uploading)
  - Registration Form
  - Forgot / Login Form
  - Upload Form
  - Dashboard Form
- Administrator (Manage documents)
  - Dashboard
  - User Management

1.2 WORKFLOW OF WEBSITE :-

Frontend User :-

1. Firstly, the User has to register by filling registration form.

2. Now, User has to login to enter into his profile.

3. After all of this, Now User can upload files and can check on dashboard his uploading files.

Admin :-

1. Here the Admin has to register by filling registration form.

2. Now the Admin can login to enter into his profile.

3. After entering into his profile Admin can see the uploaded file which was uploaded by User. Now the admin has full control to approve or reject the file or also give instructions/comments for particular changes.

1.3. TECHNOLOGIES USED :-

1.3.1 FOR FRONTEND :-

ReactJs :- Mainly technology used in this project for frontend is reactJs. React (also known as React.js) is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta and a community of individual developers and companies.

CSS :-

● CSS stands for Cascading Style Sheets.

● CSS describes how HTML elements are to be displayed on screen, paper, or in other media.

● CSS saves a lot of work. It can control the layout of multiple web pages all at once.

● External stylesheets are stored in CSS files.

HTML :-

● HTML, or Hypertext Markup Language, is a markup language for the web that defines the structure of web pages.

● It is one of the most basic building blocks of every website.

1.3.2 FOR BACKEND :-

Node.js :- Node.js (Node) is an open source development platform for executing JavaScript code server-side. Node is useful for developing applications that require a persistent connection from the browser to the server and is often used for real-time applications such as chat, news feeds and web push notifications.

FOR DATABASE :-

MySQL :-

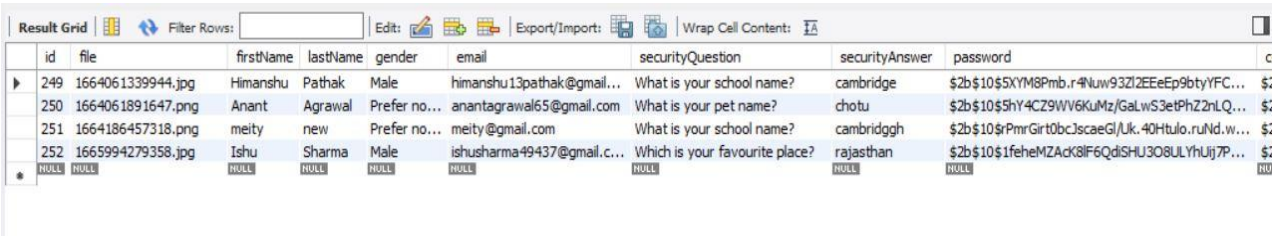
● MySQL is an open-source relational database management system. As with other relational databases, MySQL stores data in tables made up of rows and columns. Users can define, manipulate, control, and query data using Structured Query Language

● A flexible and powerful program, MySQL is the most popular open-source database system in the world. As part of the widely-used LAMP technology stack (which consists of a Linux-based operating system, the Apache web server, a MySQL database, and PHP for processing)

● It is used to store and retrieve data in a wide variety of popular applications, websites, and services.

DATA STRUCTURE OF ALL MODULES :-

STRUCTURE OF DATA REPOSITORY - PROJECT DOCUMENT SUBMISSION SYSTEM(DSS) REGISTER TABLE :- In register table, We are saving data of a particular user through registration page.



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

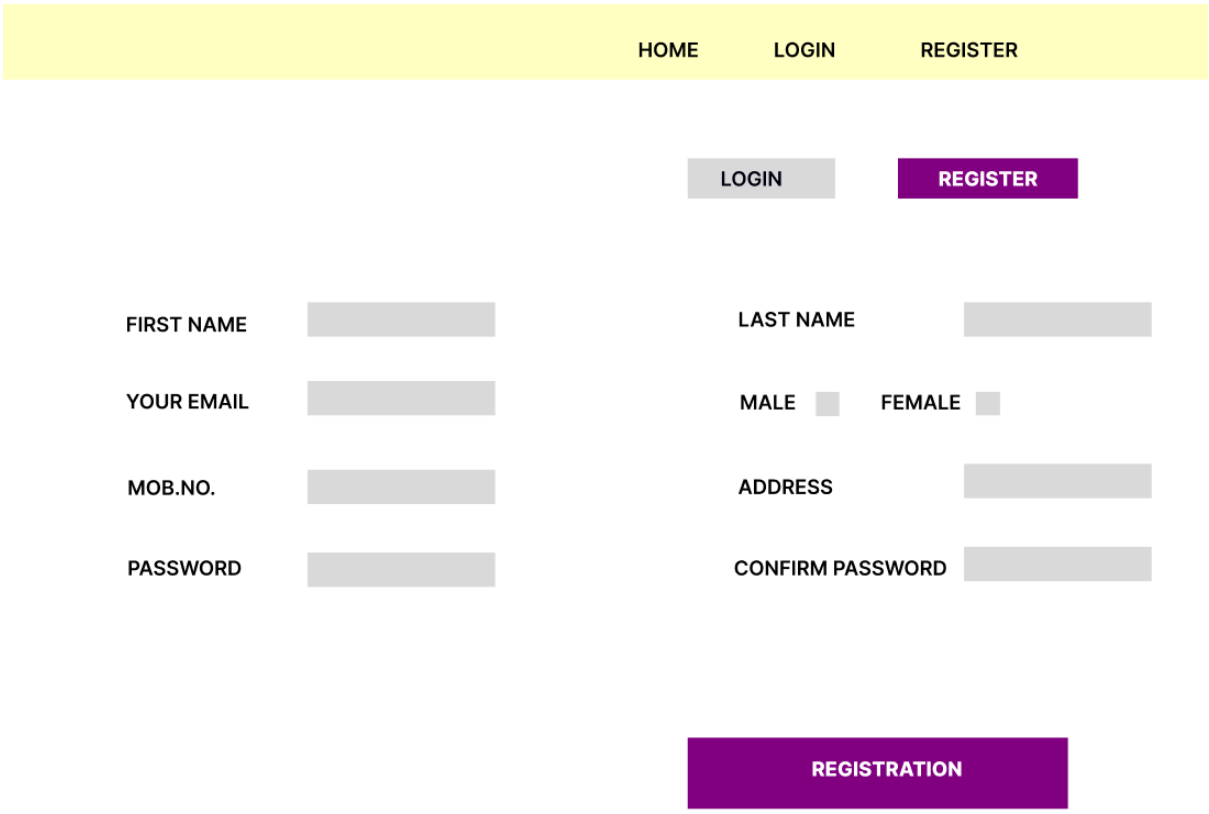
id	file	firstName	lastName	gender	email	securityQuestion	securityAnswer	password
249	1664061339944.jpg	Himanshu	Pathak	Male	himanshu13pathak@gmail...	What is your school name?	cambridge	$2b$10$5XYM8Pmb.r4Nuw93Zl2EEeEp9btyYFC...
250	1664061891647.png	Anant	Agrawal	Prefer no...	anantagrawal65@gmail.com	What is your pet name?	chotu	$2b$10$5hY4CZ9WV6KuMz/GaLwS3etPhZ2nLQ...
251	1664186457318.png	meity	new	Prefer no...	meity@gmail.com	What is your school name?	cambridggh	$2b$10$rPmrGirt0bcJscaeGl/Uk.40Htulo.ruNd.w...
252	1665994279358.jpg	Ishu	Sharma	Male	ishusharma49437@gmail.c...	Which is your favourite place?	rajasthan	$2b$10$1feheMZAcK8lF6QdiSHU3O8ULYhUij7P...
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2. COMPONENTS

● In Software Engineering, the first step is REQUIREMENT GATHERING.

● In Software Engineering, the second step is DESIGNING.

BLUEPRINT OF PPS :-



● So, we made the wireframe with the help of FIGMA in which we made the whole flow with design and prototypes of how our website will work. In figma, first we made the register page with proper utilities and navbar, after that we made the Home page and after that login page and forgot password page.

● After done with wireframe on figma. Now all this procedure is done on the VS code. With the help of react.js we make the register page with proper utilities and validation then with the help of node.js the work of backend is done and after all of this finally we connect the register page with the database where we used Mysql. All the registered fields were showing on the database.

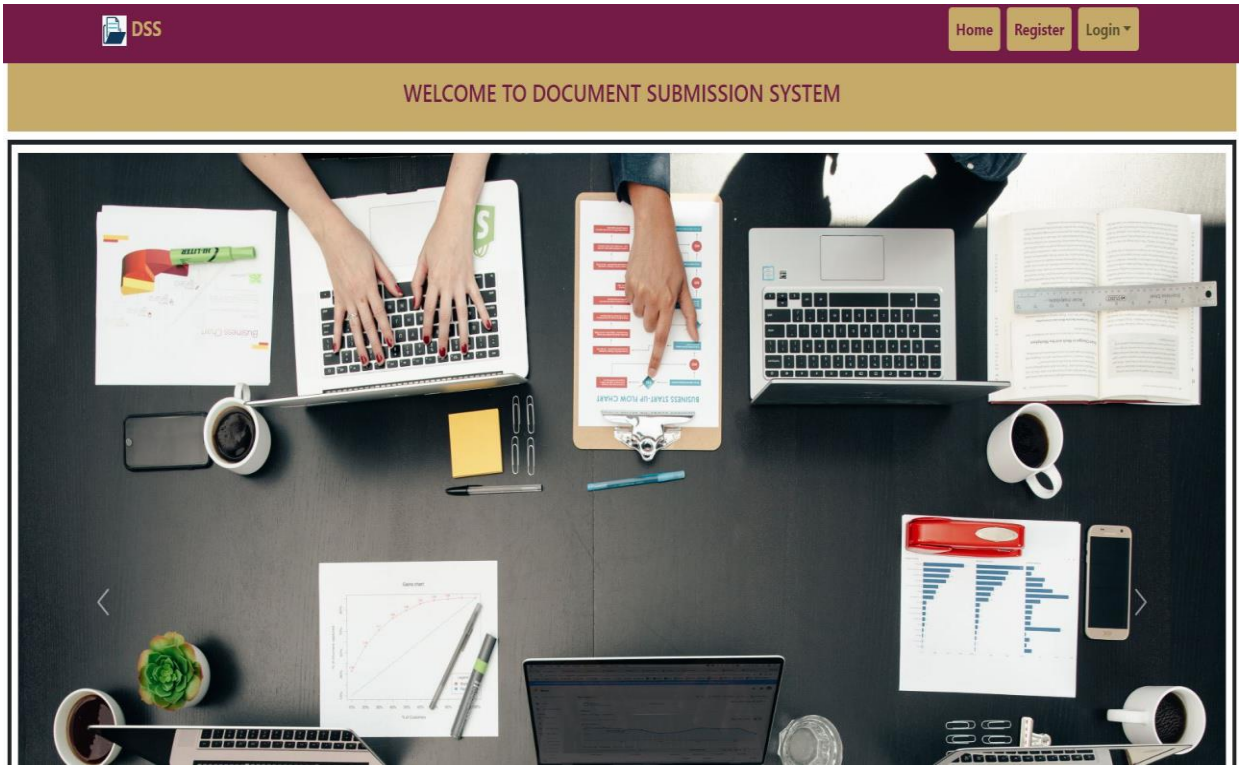
● After we finished with the Registered page, We worked on the login page and then we connected the login page with backend and database.

● Similarly, we worked on the reset page, forgot password page and the profile page. In making all the pages, react.js plays a very vital role.

2.1 HOME PAGE :-

- We added the features of Navbar.
- In Navbar, options for other pages like registration and login page are given.
- We directly navigate on the other pages by clicking on these buttons.
- Also paste the image by import feature.

PREVIEW OF HOME PAGE –



SOURCE CODE (HOME PAGE) :-

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.13.1/css/all.min.css"
integrity="sha256-2XFplPlrFClt0bIdPgpz8H7ojnk10H69xRqd9+uTShA=" crossorigin="anonymous" />
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous"></link>

<section style={{ backgroundColor: "#c5aa6a" }} className="text-center m-2 mt-2 p-2 d-flex
aligns-items-center justify-content-md-center ">

  <h4 className="p-2 mt-5 text-center pt-3" style={{ color: "#741b47" }}>WELCOME TO DOCUMENT SUBMISSION
  SYSTEM</h4>

</section>
<Carousel className='m-2 p-2 ml-2 border-5 border border-dark w-100%' >
  <Carousel.Item>
    <img
      className="w-100"
      src={image12}
      alt="First slide"
    />
    <Carousel.Caption>
      <h3 className=''>Document Submission System</h3>
    </Carousel.Caption>
  </Carousel.Item>
  <Carousel.Item>
    <img
      className="w-100"
      src={image13}
```

In above code, we use **navDropdown** function for apply the dropdown in the login button.

2.2 REGISTER PAGE :-

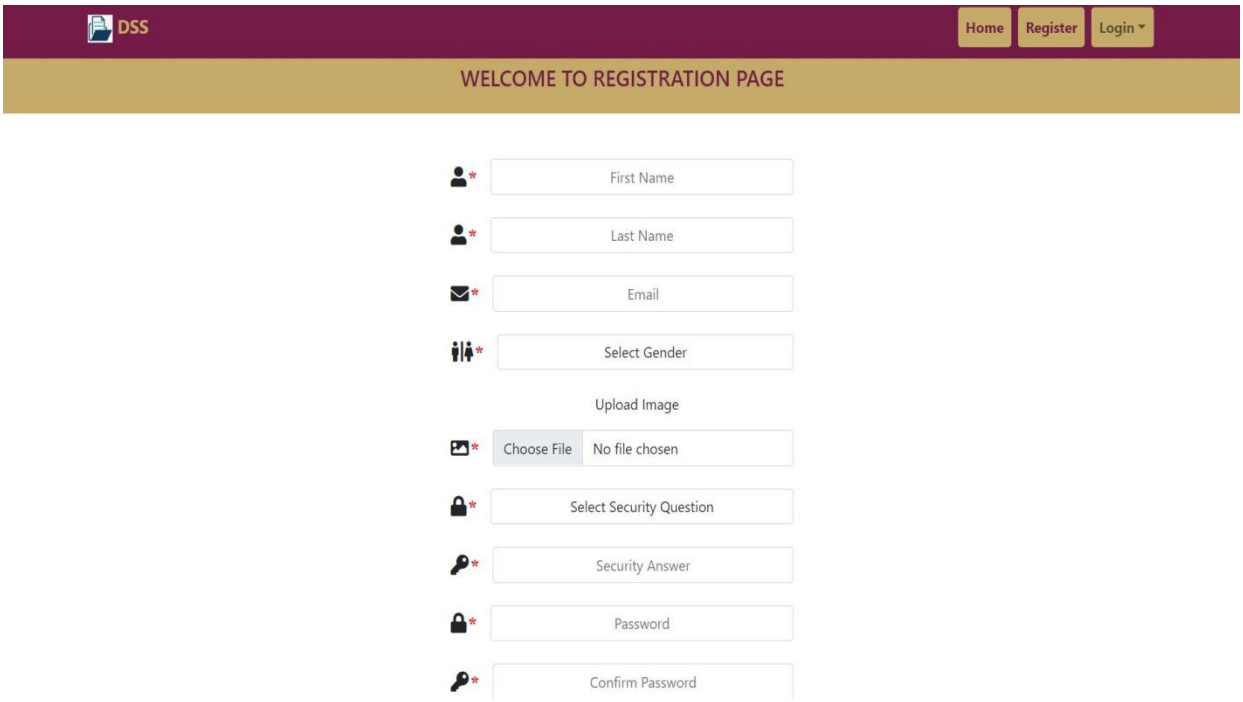
● In the Register page, all the mandatory fields are given according to requirements.

● We have used validations like regex validators for email validation and password validations.

● Passwords will be saved in encrypted format in storage.

● Upload image field is also there with acceptance of only jpg/jpeg/png format.

PREVIEW OF REGISTER PAGE:-



SOURCE CODE (REGISTER PAGE) :-

```
<div class="container mt-5 ">
    <div className="row aligns-items-center justify-content-center">
        <div className="col-sm-4">
            <form onSubmit={handleSubmit}>


                <div className="d-flex flex-row align-items-center mb-4 ">
                    <MDBIcon fas icon="user me-3" size='lg' className='required-field'/>
                    <input
                        className='form-control text-center'
                        type="text"
                        name="firstName"
                        placeholder="First Name"
                        value={formValues.firstName}
                        onChange={handleChange} />
                </div>
                {formErrors && !formValues.firstName && <p className='text-center alert-danger'>First
                Name is required</p>}
```

● Firstly, the user page is made by the use of react js with proper validation and functionality. After this, connection of backend and database is done with the help of making API.

● After clicking on the register button the API will fetch all the data and store that data in the database.

● So the credential which will be submitted by the user will be saved in the database.

2.3 LOGIN PAGE :- There is a drop down in the login button. From where we choose the category of login page like:-

1. User login page

2. Admin login page

2.3.1 FOR USER LOGIN :-

● In the User login page, all the mandatory fields are given according to requirements. Passwords will be compared in decrypted order from storage.

● If the email is already registered it will redirect you to PROFILE PAGE.

● When the user will fill credentials, an API works behind the backend and fetches the credential which is filled by the user with the registered credential and if both the credentials are the same. User will authorize to login.

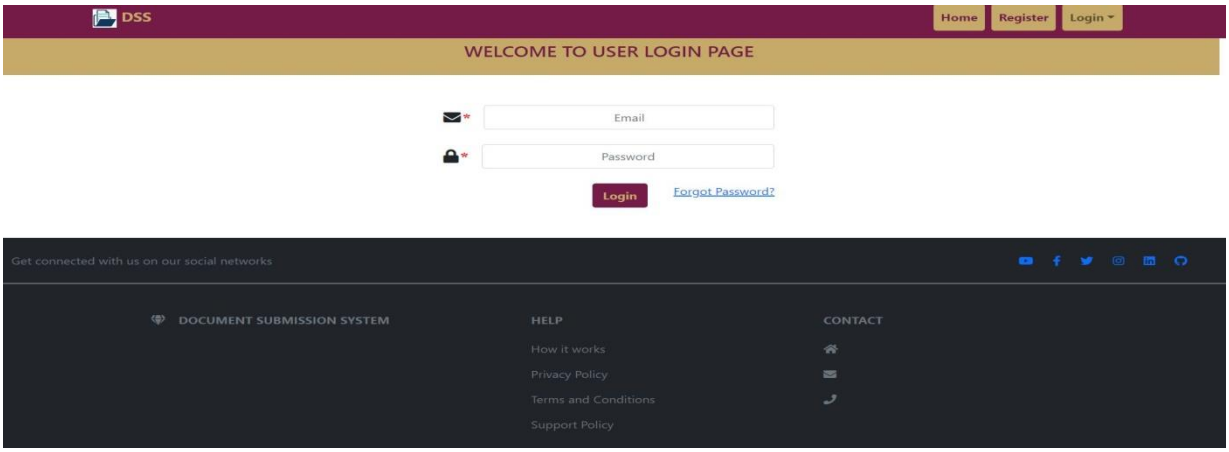
● If the credentials are not the same then the alert message will be prompted.

● After login successfully, the user will come on the profile page where the image, user's name, gender and the email address will show on the screen.The navBar is also changed.

● Profile, dashboard, upload file and logout button will show in the navBar.In the Dashboard page, Users will be able to see the uploaded file.

● There is an option to upload files in the navBar, after clicking on this option, the user will render on the page where he/she will choose the intended file and also attach a message with the file and then upload.

PREVIEW OF USER LOGIN PAGE:-



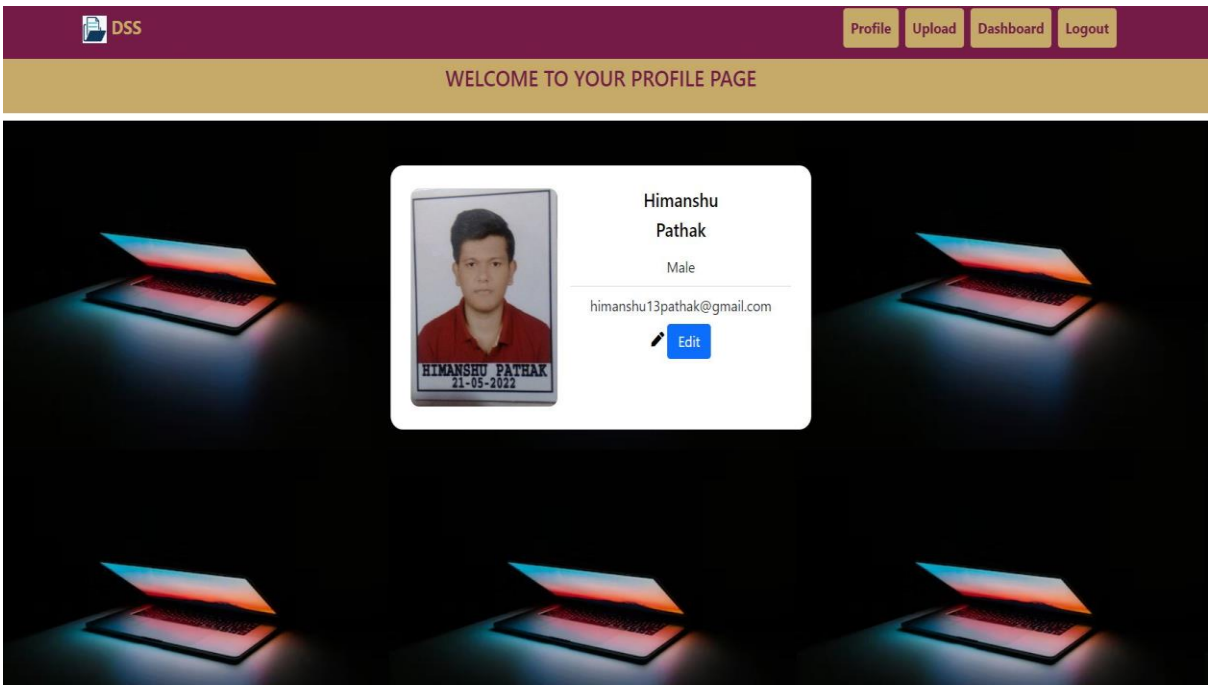
SOURCE CODE(USER LOGIN PAGE):-

```
            <div className="d-flex flex-row align-items-center mb-4 ">
                <MDBIcon fas icon="lock me-3" size='lg' className='required-field'/>
                <input
                    className='form-control text-center'
                    type="password"
                    name="password"
                    placeholder="Password"
                    value={formValues.password}
                    onChange={handleChange} />
            </div>
            <p className='text-center justify-content alert-danger'>{formErrors.password}</p>

            <div className="row justify-content-end">
                <div className='col-3'>
                    <button className="btn ml-2"style={{backgroundColor:"#741b47",color:"#c5aa6a",
                    fontWeight: 'bold'}}>Login</button>
                </div>
                <div className='col-auto'>
                    <Link to="/Forget" className="">Forgot Password?</Link>
                </div>

            </div>
        </form>
    </div>
</div>
```

PROFILE PAGE :-

● In the profile page, User's information will be shown like Image, name, gender and Email Id.

PREVIEW OF USER PROFILE :-



SOURCE CODE(USER PROFILE PAGE):-

```
<div className="vh-100" style={{ backgroundImage: `url(${image16})` }}>
    <MDBContainer>
        <MDBRow className="justify-content-center">
            <MDBCol md="9" lg="7" xl="5" className="mt-5">
                <MDBCard style={{ borderRadius: '15px' }}>
                    <MDBCardBody className="p-4">
                        <div className="d-flex text-black">
                            <div className="flex-shrink-0">
                                <MDBCardImage
                                    style={{ width: '180px', borderRadius: '10px' }}
                                    src={src}
                                    alt='Generic placeholder image'
                                    fluid />
                            </div>
                            <div className="flex-grow-1 ms-3">
                                <MDBCardTitle>{JSON.parse(auth).firstName}</MDBCardTitle>
                                <MDBCardTitle>{JSON.parse(auth).lastName}</MDBCardTitle>
                                <ListGroup className="list-group-flush">
                                    <ListGroup.Item >{JSON.parse(auth).gender}</ListGroup.Item>
                                    <ListGroup.Item >{JSON.parse(auth).email}</ListGroup.Item>
                                </ListGroup>
                                <div className="text-center">
                                    <MDBIcon fas icon="pen me-1" />
                                    <MDBBtn className="flex-grow-1">Edit</MDBBtn>
                                </div>
```

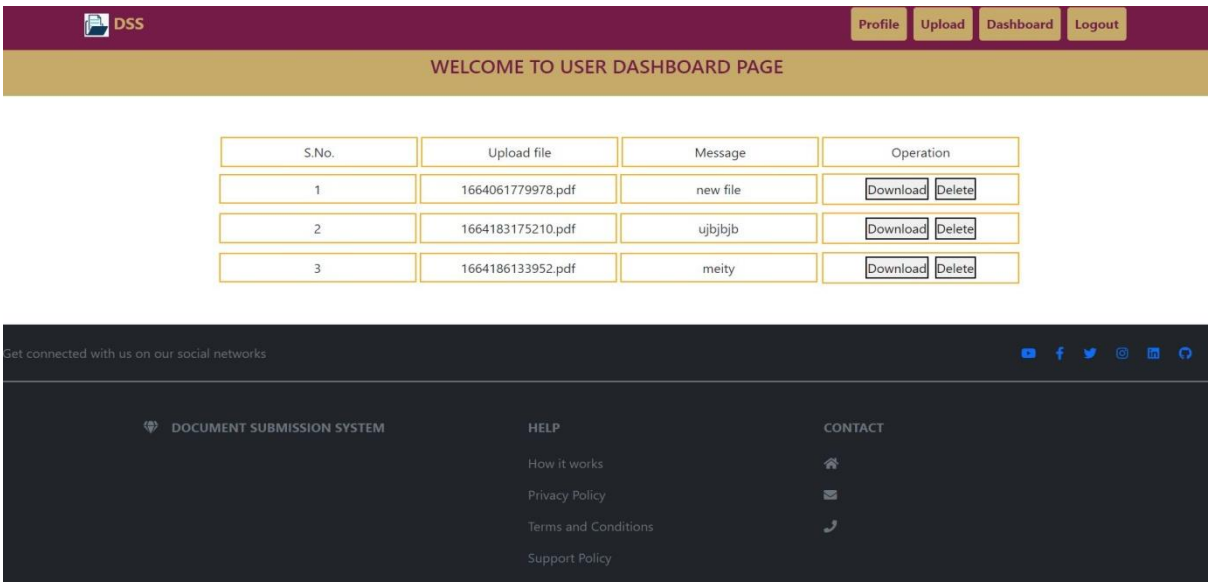
USER DASHBOARD :-

● In the dashboard, different types of fields are given.

● Any file which will be uploaded by the user will show in the dashboard with an attached message.

● Users can also view the uploaded file from the operation field.

PREVIEW OF USER DASHBOARD PAGE:-



SOURCE CODE (USER DASHBOARD PAGE) :-

```
<div className='product-list'>
  <ul >
    <li>S.No.</li>
    <li>Upload file</li>
    <li>Message</li>
    <li>Operation</li>
  </ul>
  {
    products.length > 0 ? products.map((items, index) =>
      <ul>
        <li>{index + 1}</li>
        <li>{items.uploadfile}</li>
        <li>{items.message}</li>
        <li className='m-1 p-0'>
          <button onClick={(e)=>dodownload} className='m-1 p-0'>Download</button>
          <button onClick={(e)=>doDelete} className='m-1 p-0'>Delete</button>
        </li>
      </ul>
    )
      : <h1>No Result Found</h1>
  }
```

2.3.2 FOR ADMIN LOGIN :-

● After clicking on the admin in the drop down. Admin login page is rendered on the screen.

● This page is only for admin to login. Admin has a unique username and password

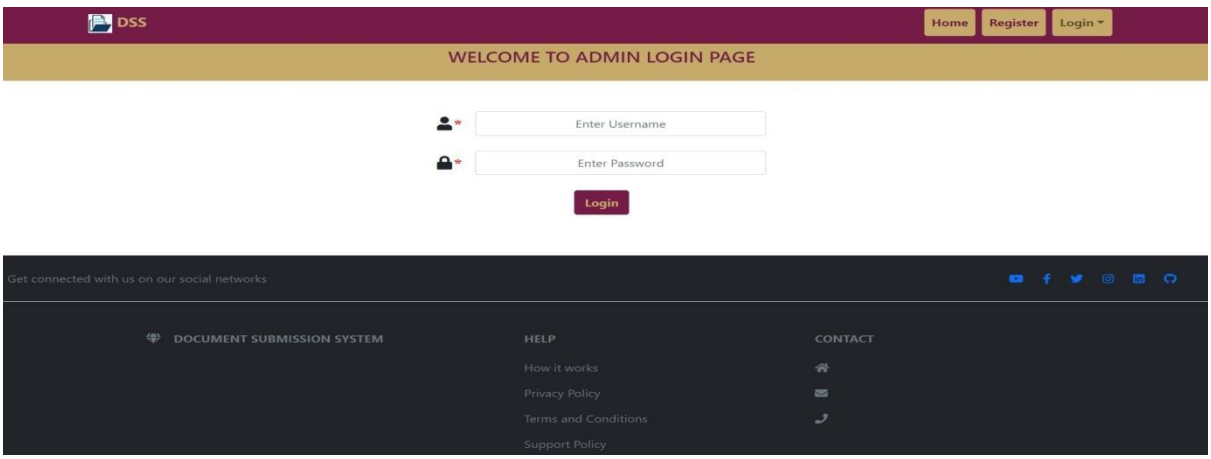
● Without a unique username and password the admin is not able to login.

● The username and password will be predefined to the admin and will store in the database.

● After successful login, admin will come on the profile page. Profile, dashboard and logout buttons are shown in the navBar.

● On clicking the dashboard button, admin has access to see the files which will be uploaded by the user.

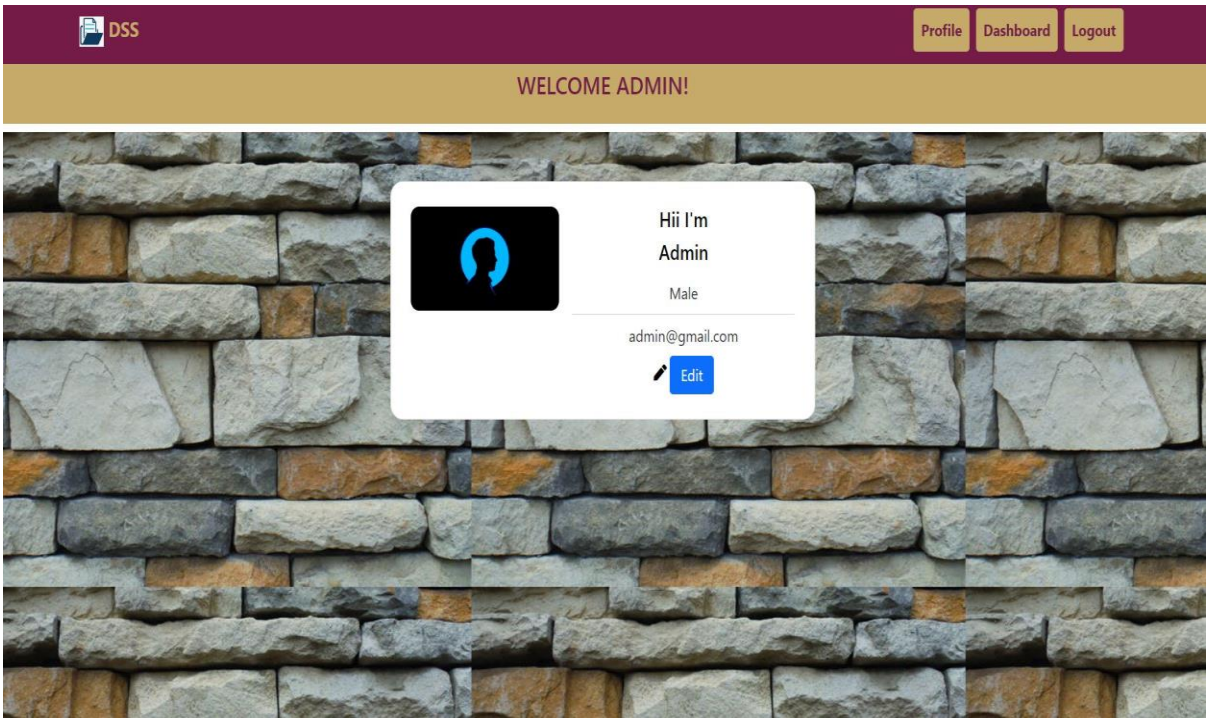
PREVIEW OF ADMIN LOGIN PAGE –



ADMIN PROFILE :-

● In the profile page, Admin's information will be shown.

PREVIEW OF ADMIN PROFILE PAGE:-



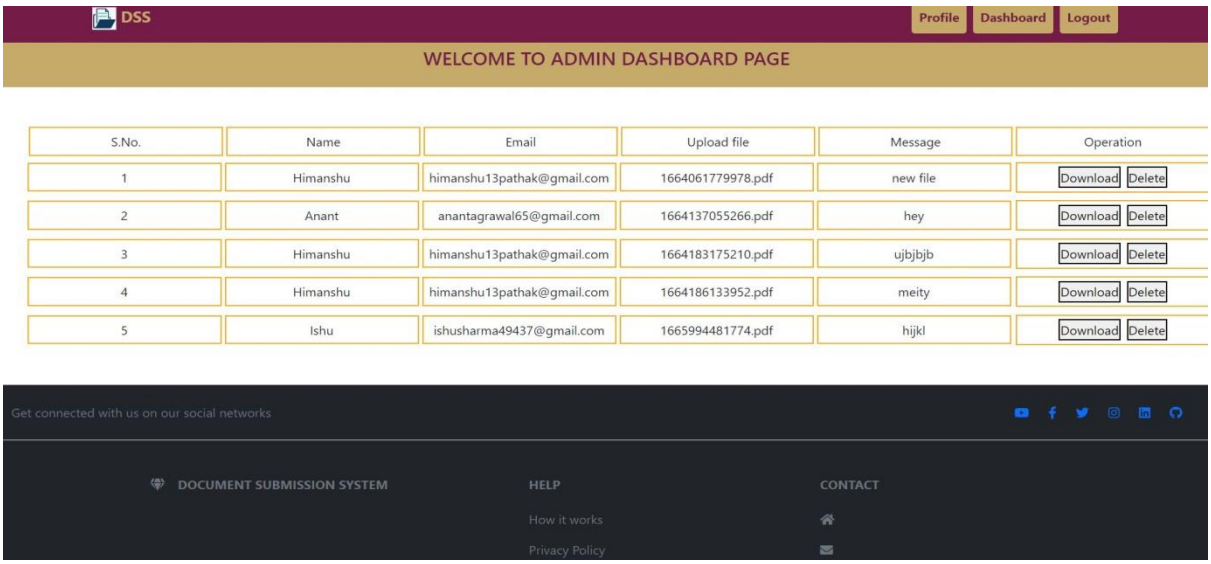
ADMIN DASHBOARD :-

● In the dashboard, different types of fields are given.

● Any file which will be uploaded by the user will show in the dashboard of admin with an attached message with proper username.

● Admin take actions like view the file or download the file.

PREVIEW OF ADMIN DASHBOARD PAGE:-

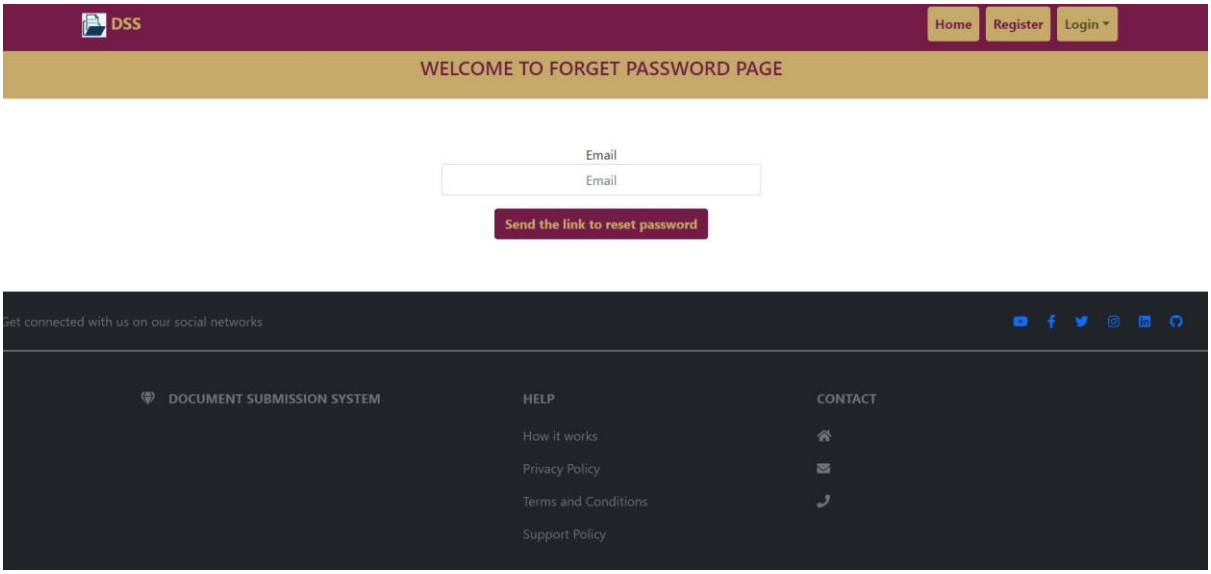


FORGOT PAGE :-

● In the forgot page, there is a field given for entering email.

● After entering the email and clicking on the “send the link to reset password”, a unique link will be sent to your registered email ID.

PREVIEW OF FORGOT PAGE –



● This link will be valid for only 15 minutes.

● On clicking the unique reset password link, the user will render to the reset password page.

● When the user successfully enters Password and Confirm Password same, the password will be changed successfully and saved in the database in hashed form.

● Here we have made the API for the forgot page, when any user clicks on the forgot password, the forgot password page will be rendered on the screen.

● After hitting the reset password API hits and checks the email id from the database and sends the reset link to the given email address.

SOURCE CODE (FORGOT PAGE) :-

```
<div class="container mt-5 ">
    <div className="row aligns-items-center justify-content-center">
        <div className="col-sm-4">
            <form onSubmit={handleSubmit}>
                <div className="field mb-3 m-2 text-center form-group">
                    <label>Email</label>
                    <input
                        className='form-control text-center'
                        type="email"
                        name="email"
                        placeholder="Email"
                        value={formValues.email}
                        onChange={handleChange} />
                </div>
                <p>{formErrors.email}</p>

                <div className='mb-3 m-2 text-center form-group'>
                    <button className="btn text-center"style={{backgroundColor:"#741b47",
                    color:"#c5aa6a",fontWeight: 'bold'}}>Send the link to reset password</button>
                </div>
```

RESET PAGE :-

● The reset page is created with the use of ejs.

● As we could see in the forgot page, the unique reset link will be sent on the given email address.

● After clicking on the link, the reset page renders on the screen

● When the user will click on the reset password button after filling all the fields. The user's new password will save in the database in place of the old password with proper Hashed format.

● After this, Screen will navigate to the login page.

● To encrypt the password, a Bcrypt function is used. In which salt is used for generating hash passwords.

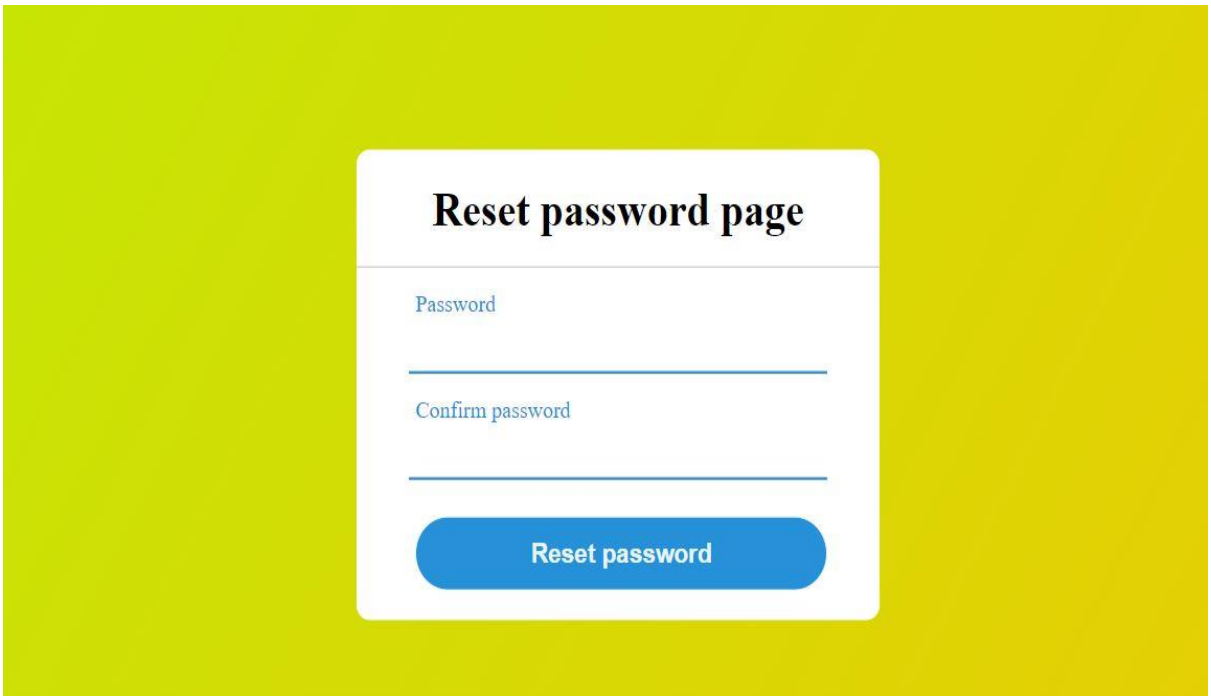
● Rendering from one page to another page is done with the help of navigation function.

● The Unique Reset password Link is created through Jsonwebtoken (JWT) and Unique Email Id.

● This JWT token is valid only for 15 min and this unique Reset password link is sent to a unique email I'd through Googleapis and Nodemailer.

● For this, We have used Unique REFRESH_TOKEN created on Google Cloud Console for a particular Web-Client, and By calling Asynchronous sendMail function, a unique link sent successfully to particular email I'd.

PREVIEW OF RESET PAGE –



SOURCE CODE (RESET PAGE) :-

```
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Reset password page</title>
    <script>
        var status = '<%- JSON.stringify(status) %>';
        if (JSON.parse(status) == "verified") {
            if (confirm("Do you want to login?")) {
                window.location.href = "http://localhost:3002/UserLogin";
            }
        }
    </script>
    <style>
        body {
            margin: 0;
            padding: 0;
            font-family: montserrat;
            background: linear-gradient(120deg, #bdeb05, #efc804);
            height: 100vh;
            overflow: hidden;
        }
```

CONCLUSION & FUTURE WORK:-

1. We can use this project also for resume submission for placements in colleges.

2. We can develop it with some more functionalities given to admin.

3. We will ensure this project to work in many colleges for resume submission system in the future.

4. We will add more UI friendly interfaces.

5. We will host this website.

REFERENCES:-

1. David Herron (2014) 'Node.js web development', No.1, pp.201-219.
2. Simon Holmes(2017) 'Getting MEAN with Mongo, Express, Angular, and Node', pp.67–89.
3. Code step by step, 'Nodejs tutorials', youtube channel.