

MEMORY VAULT

A PROJECT REPORT

Submitted by

JERIN BS (210701095)

in partial fulfillment for the course

CS19542 - INTERNET PROGRAMMING

*for the award of the degree
of*

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING



RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

NOVEMBER 2023

BONAFIDE CERTIFICATE

Certified that this project “**MEMORY VAULT**” is the bonafide work of “**JERIN B S**” who carried out the project work under my supervision.

SIGNATURE

Dr. P. KUMAR M.E., Ph.D.,
HEAD OF THE DEPARTMENT

Professor and Head
Dept. of Computer Science and
Engg,
Rajalakshmi Engineering College
Chennai.

SIGNATURE

Ms. K. DEEPAK KUMAR, M.E.,
SUPERVISOR

Assistant Professor (SS)
Dept. of Computer Science and
Engg,
Rajalakshmi Engineering College
Chennai.

Submitted to Project and Viva Voce Examination for the subject CS19542 -

Internet Programming held on _____.

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

As the world gets leveraged towards a digital life, people tend to capture the memories of each moment digitally rather than merely storing it in their brain cells. As the amount of data getting generated each day increases, the organization of data grows to be a matter of concern.

This project aims to develop a robust and user-friendly web application that facilitates secure user authentication, daily memory entry creation, and seamless interaction with memories. The system includes a feature-rich text editor and image upload functionality for creating memories, along with location tagging for associating memories with specific places. The timeline view allows users to scroll through their memories chronologically, providing a visual preview of text, images, and location tags. Additionally, the application incorporates search functionality with filters for date, location, and tags, enhancing personalized memory exploration.

To ensure data security, the system implements secure transmission of data between the client and server. Users can edit and update existing memory entries, and a secure deletion mechanism is in place for removing unwanted memories. Reminders can be set to encourage regular memory entry, and push notifications serve as reminders for users to capture their daily memories.

The collaborative aspect of the app allows users to share memories with friends, fostering a social component. Through this comprehensive approach, the application aims to provide a secure, interactive, and enjoyable platform for users to capture, explore, and share their memories.

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honorable chairman **Mr.MEGANATHAN S** and the chairperson **Dr. THANGAM MEGANATHAN M** for their timely support and encouragement.

We are greatly indebted to our respected and honorable principal **Dr. MURUGESAN S N** for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by our Head of the Department **Dr. KUMAR P M.E Ph.D.**, for being ever supporting force during our project work.

We also extend our sincere and hearty thanks to our internal guide **Mr. DEEPAK KUMAR K**, for his valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of computer science engineering.

JERIN BS (210701095)

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	(ii)
I	INTRODUCTION	
	1.1 DESCRIPTION	1
	1.2 SCOPE OF WORK	4
	1.3 PROBLEM STATEMENT	5
	1.4 AIM & OBJECTIVE	6
II	SYSTEM SPECIFICATIONS	
	2.1 HARDWARE SPECIFICATIONS	8
	2.2 SOFTWARE SPECIFICATIONS	8
III	PROJECT OVERVIEW	
	3.1 FRONTEND	9
	3.2 BACKEND	9
	3.3 API	10
	3.4 SECURITY ASPECTS	10
IV	PROJECT DIRECTORY STRUCTURE	11
V	DATABASE DESIGN	12
VI	MODULE DESCRIPTION	
	6.1 LOGIN PAGE	13
	6.2 MEMORIES PAGE	13
	6.3 ADD MEMORY PAGE	13
VII	SYSTEM DESIGN	
	7.1 SYSTEM FLOW DIAGRAM	14
VIII	SOURCE CODE	
	8.1 APP.JS	16
	8.2 MEMORIES PAGE	17
	8.3 LOGIN PAGE	19
	8.4 MEMORY CARD COMPONENT	22
	8.5 SERVER.JS	25
	8.6 MEMORY ROUTE	26
	8.6 MEMORY MODEL	28
IX	EXECUTION	29
X	CONCLUSION & FUTURE ENHANCEMENTS	38
XI	REFERENCES	39

LIST OF FIGURES

Figure No.	Figure Description	Page No.
1	Directory Structure	11
2	Database Structure	12
3	Flow diagram	15
4	Login - Raw	29
5	Forgot Password	29
6	Forgot Password - With details	30
7	Reset Password	31
8	Login - With Details	31
9	Memories	32
10	Memory	32
11	Add Memory	33
12	Add Memory - With Details	33
13	Memories - After Adding	34
14	Memory - To Be Deleted	34
15	Memories - After Deleting	35
16	Editing A Memory	35
17	User Settings	36
18	Login - Another Account	36
19	Memories - Another User	37

LIST OF ABBREVIATIONS

ABBREVIATION	DEFINITION
IP	Internet Protocol
API	Application Programming Interface
CRM	Customer Relationship Management
ERP	Enterprise Resource Planning
IDE	Integrated Development Environment
UML	Unified Modeling Language
UI	User Interface

CHAPTER I

INTRODUCTION

1.1 DESCRIPTION

In an era characterized by the ubiquitous presence of digital devices, our lives are increasingly interwoven with technology, shaping the way we perceive, remember, and share our experiences. The evolution from analog to digital has not only transformed the means by which we capture and document our memories but has also given rise to an unprecedented influx of data. As each moment is translated into pixels and stored in the vast expanses of digital repositories, the organization and meaningful curation of this colossal dataset have emerged as critical challenges. Recognizing this paradigm shift, our project embarks on the development of a robust and user-friendly web application poised to serve as a digital sanctuary for memories – a dynamic space that transcends mere storage, fostering interaction, security, and a sense of collective nostalgia.

At its core, this initiative seeks to address the burgeoning need for a comprehensive solution that not only accommodates the escalating volume of digital memories but also enriches the very act of preserving and revisiting these moments. The envisioned web application is positioned as a multifaceted platform that integrates advanced functionalities with user-centric design principles, ensuring that the process of capturing, exploring, and sharing memories becomes a seamless and enriching experience.

Central to the application's design is the recognition that memories are multifaceted – comprising not only textual descriptions but also the rich

tapestry of images that encapsulate the essence of a moment. Thus, the system boasts a feature-rich text editor complemented by an intuitive image upload functionality. This combination empowers users to encapsulate the full spectrum of their experiences, allowing for a nuanced and vivid representation of memories within the digital realm.

Moreover, the integration of location tagging introduces a spatial dimension to the memories, enabling users to associate their experiences with specific places. In doing so, the application not only preserves the temporal aspect of memories but also anchors them in the physical world, creating a nuanced and holistic narrative of one's journey through time and space. This layer of contextualization adds depth and resonance to the digital memories, bridging the gap between the virtual and the lived reality.

Navigating this expansive repository of memories is made intuitive and engaging through the implementation of a timeline view. This chronological arrangement of entries provides users with a visual journey through their past, offering glimpses of text, images, and location tags. The timeline serves as a dynamic canvas, allowing individuals to traverse the landscape of their memories in a manner that mirrors the organic flow of time.

To further enhance the user experience, the application incorporates robust search functionality coupled with filters for date, location, and tags. This strategic amalgamation of search and filtering mechanisms amplifies the personalization of memory exploration, empowering users to tailor their journey through the digital chronicles according to their preferences and moods. This nuanced approach recognizes the diversity inherent in the human experience, ensuring that the application is not a

static vault but a dynamic and responsive companion in the exploration of one's memories.

Crucial to the success of this digital repository is the unwavering commitment to data security. In an age where privacy concerns loom large, the system employs state-of-the-art encryption and secure transmission protocols to safeguard user data during transit between the client and server. Users are bestowed with a sense of control over their digital archives, with the ability to edit, update, or securely delete existing memory entries. This commitment to privacy and user autonomy stands as a testament to our project's dedication to fostering a secure and trustworthy digital environment.

Beyond the individual experience, the web application acknowledges the inherently social nature of memory-sharing. Memories are not static entities confined to the individual; they are dynamic narratives that gain depth and meaning when shared with others. To cater to this communal aspect, the application introduces a collaborative dimension, enabling users to share their memories with friends and loved ones. This infusion of a social component transforms the act of memory-sharing into a collective experience, fostering connections and shared narratives.

Recognizing the challenges posed by the rapid pace of modern life, the application incorporates reminders and push notifications. These gentle nudges serve as a catalyst for users to engage with the platform regularly, ensuring that the act of memory creation becomes a consistent and integral part of their daily routine. This not only contributes to the longevity of the digital memories but also underscores the application's commitment to becoming an active and participatory component of the user's life.

In essence, this comprehensive web application aspires to redefine the landscape of digital memory management. It is more than a repository; it is a living, breathing space that mirrors the complexity, diversity, and interconnectedness of human experiences. By seamlessly blending security, interactivity, and social connectivity, the application endeavors to provide a secure, user-friendly, and enjoyable platform. It is a testament to the transformative power of technology when harnessed with a human-centric approach – a gateway to the digital realm where memories are not just stored but actively woven into the fabric of our evolving narratives.

1.2 SCOPE OF WORK

This project is envisioned to cater to a diverse spectrum of users seeking a sophisticated yet user-friendly platform to capture, curate, and share their digital memories. The primary intended users include individuals of all age groups who embrace the digital lifestyle and aspire to chronicle their life experiences in a dynamic and secure manner. This application is tailored for those who find value in seamlessly blending textual descriptions, images, and location data to construct a comprehensive narrative of their personal journey.

The platform accommodates the tech-savvy millennial generation and beyond, providing an intuitive and interactive space for them to document and revisit their memories. Professionals, students, and creative enthusiasts seeking a holistic digital memory management solution will find this application instrumental in organizing and expressing their life stories.

Furthermore, the application recognizes the importance of social connectivity in memory-sharing. It appeals to users who value the communal aspect of digital interaction, allowing them to effortlessly share their memories with friends and family. Whether for personal reflection, collaboration, or sharing life's moments, this application extends its scope to embrace a wide demographic, ensuring that individuals from various walks of life can engage with their digital memories in a meaningful and enjoyable manner.

1.3 PROBLEM STATEMENT

In an era dominated by digital documentation, the exponential growth in data generated from daily experiences has presented a pressing challenge – the effective organization and meaningful interaction with this vast reservoir of memories. As individuals increasingly shift towards capturing their life moments in digital formats, the absence of a comprehensive and user-friendly platform has become apparent. Existing solutions often fall short in providing an integrated system that seamlessly combines secure user authentication, multi-faceted memory entry creation, and dynamic interaction with memories.

The absence of a robust digital memory management solution is acutely felt by users who seek a secure, intuitive, and feature-rich platform. The current landscape lacks a unified approach that accommodates the diverse elements of memories – textual descriptions, images, and location data – while ensuring privacy and data security. Additionally, the lack of an engaging timeline view, coupled with advanced search and filter

functionalities, hinders users from exploring their digital memories in a personalized and holistic manner.

Furthermore, the dearth of a collaborative dimension in existing platforms overlooks the inherently social nature of memory-sharing. The absence of reminders and push notifications also contributes to a sporadic engagement with memory creation, diminishing the potential for establishing consistent and meaningful digital narratives.

In addressing these gaps, our project aims to provide a solution that not only captures the essence of digital memories but also elevates the user experience through a holistic, secure, and interactive approach. By recognizing the evolving needs of users in the digital age, this project endeavors to bridge the existing gaps in digital memory management, offering a transformative and enjoyable platform for individuals to chronicle, explore, and share the mosaic of their lives.

1.4 AIM & OBJECTIVE

The aim of this project is to develop a comprehensive and user-centric web application that redefines the digital memory management experience. The primary objective is to provide a secure and intuitive platform for users to seamlessly capture, explore, and share their memories in a multifaceted manner. The application aims to amalgamate advanced features, including a robust text editor, image upload functionality, and location tagging, fostering a dynamic representation of personal narratives. The timeline view and search functionalities aim to enhance the user's ability to navigate and personalize their memory exploration. Ensuring data security is a paramount objective, with secure

transmission protocols, user-controlled editing, and deletion mechanisms in place. The project also seeks to incorporate a collaborative dimension, allowing users to share memories with their social circles. Ultimately, the objective is to create an engaging, user-friendly, and secure digital environment that becomes an integral part of users' lives, encouraging consistent and meaningful interaction with their digital memories.

CHAPTER II

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS

Processor : i5 (2.1 GHz or higher) / M1 Chip
RAM : 4 GB 4800MHz (Minimum)
STORAGE : 10 GB (Minimum)

2.2 SOFTWARE SPECIFICATIONS

Operating System : WINDOWS 8 (or higher)
Architecture : x32 bit / x64 bit
Software Stack : Node js, Mongo Client (any)
File Formats : JS, HTML, CSS, JSX, JSON, ICO, ENV, TXT

CHAPTER III

PROJECT OVERVIEW

DESIGN STRATEGY	: Model View Controller (MVC)
ARCHITECTURAL PATTERN	: Client – Server Architecture

3.1 FRONTEND :

The client side uses JavaScript, HTML, CSS for views. React js is used for rendering the views onto the browser. All the views follow the Material UI representation recommended by Alphabet Inc. Google. The different pages are accessed through Routes and Protected Routes for security.

PAGES : Login Page, Forgot Password Page, Reset Password Page, Request OTP Page, Sign Up Page, Memories Page, Add Memory Page

3.2 BACKEND :

The Backend is a No SQL Database – Mongo DB. All data are stored in JSON document format. The image files are stored as it is with multer package in the server. CRUD (Create, Read, Update, Delete) operations are properly configured with the DB.

3.3 APPLICATION PROGRAMMING INTERFACE (API) :

The project uses RESTful API with Express js. Mongoose package is used to create object data locally and uploaded in the server. API is configured to accept only GET and POST requests. API runs on Node js environment.

NO OF END POINTS : 8

3.4 SECURITY ASPECTS :

Authentication is done by JSON Web Tokens (JWT) and the sessions are maintained using Cookie Data. Cross – Origin Resource Sharing (CORS) is configured with only client Origin. Every Password is hashed before being stored in Database. 6 out of 8 endpoints pass through JWT verification. No endpoint is static to be accessed via URL. Every Data is sent only via response and not by any other means.

CHAPTER IV

PROJECT DIRECTORY STRUCTURE

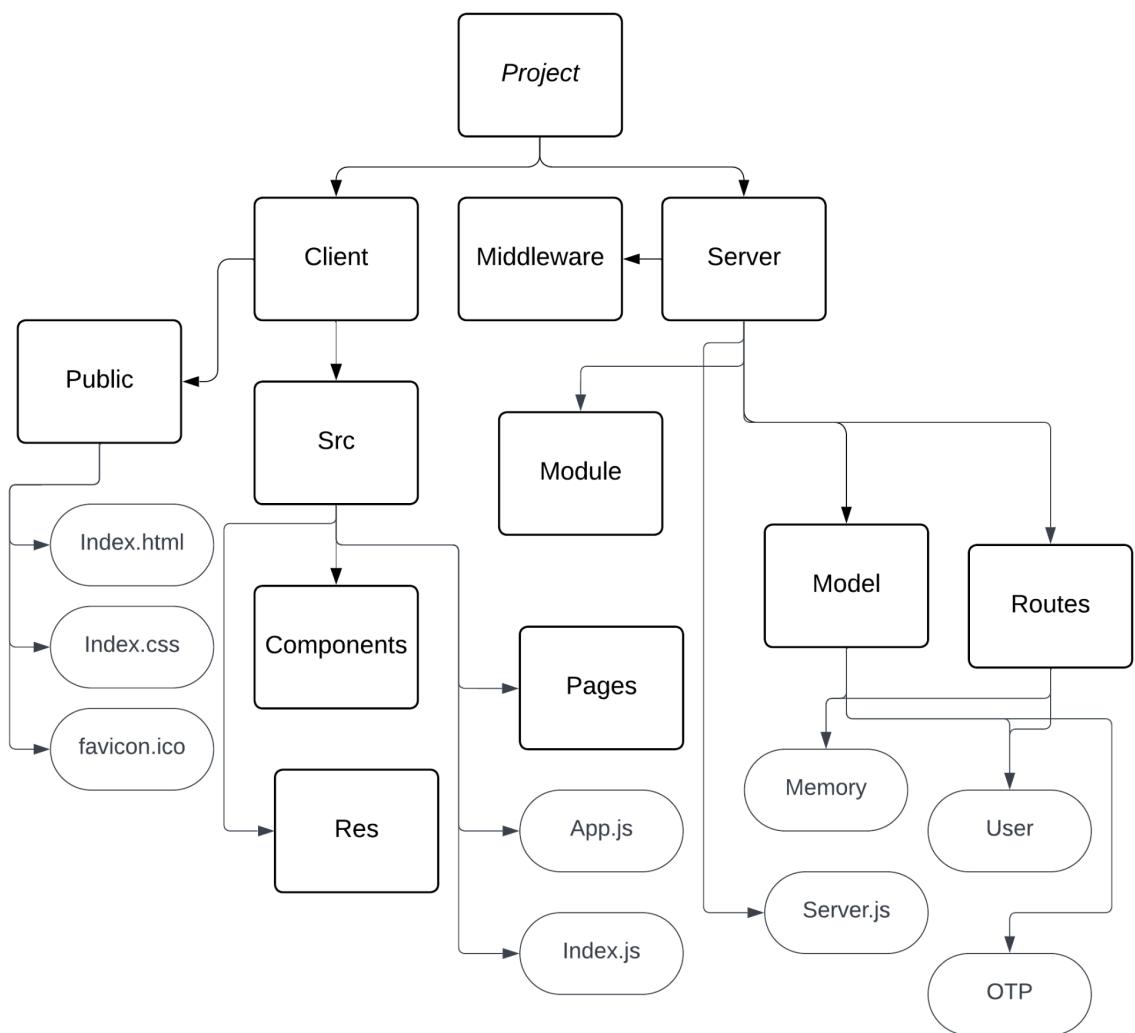


Fig.1 Directory Structure

CHAPTER V

DATABASE DESIGN

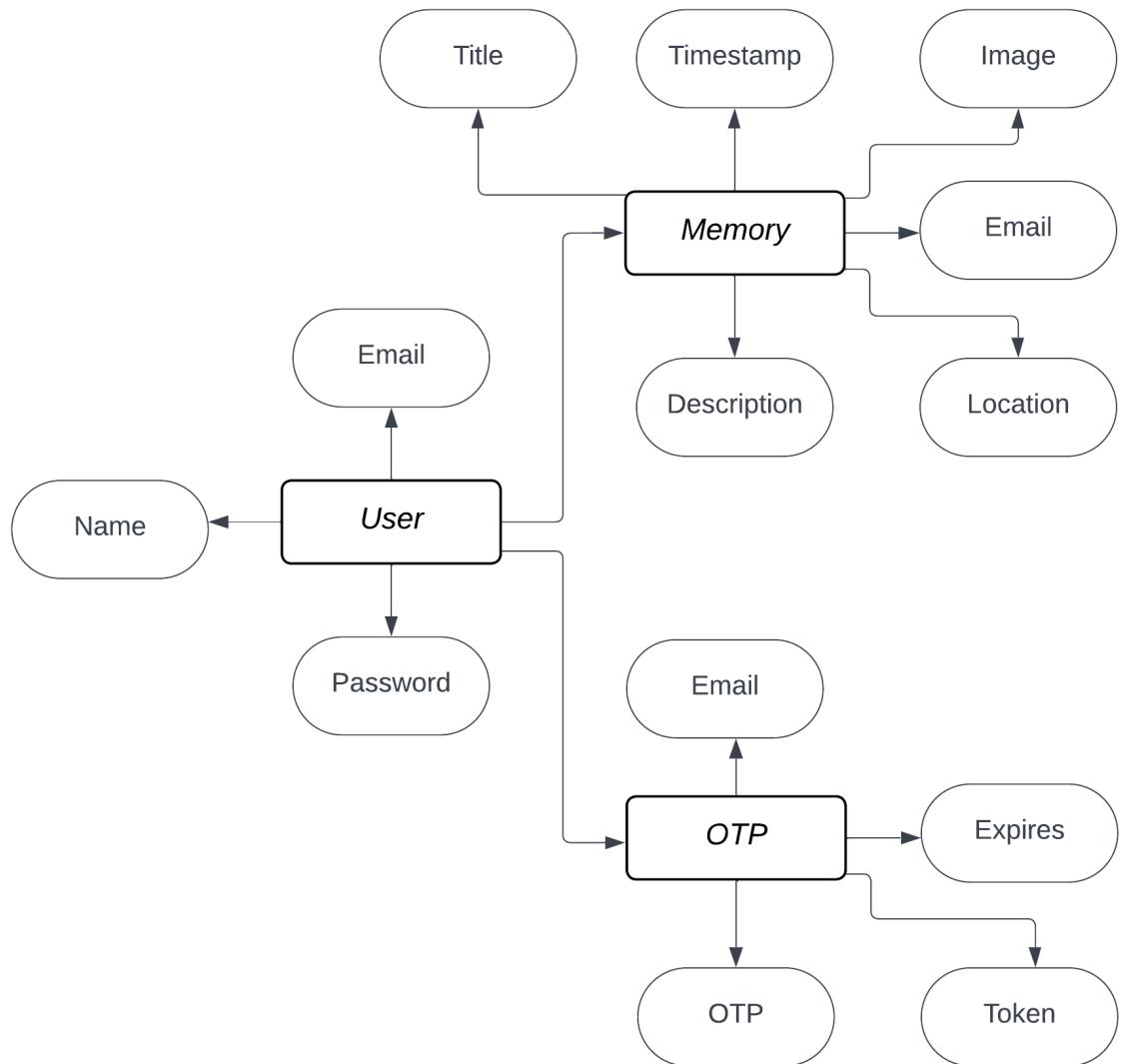


Fig.2 Database Structure

CHAPTER VI

MODULE DESCRIPTION

The project consists of 3 modules – Login Page, Memories Page and Add Memory Page. Upon start, the user will be granted with Login Page. Depending on the input, the subsequent pages will be rendered.

6.1 LOGIN PAGE :

Default Page with no authentication to access. Consists of 2 inputs – Email and password . Depending on the combination of inputs, the user will be redirected to the next page.

6.2 MEMORIES PAGE :

Requires authentication and can be accessed only by the respective student. Contains 3 components – a navigation bar, Memories in cards and Footer. The navigation bar will contain the name of the user, profile, navigation button. The cards shall contain the Memory details previously applied by the user which can be edited or deleted.

6.3 ADD MEMORY PAGE :

Contains 4 inputs – Memory Title, Description, Location and File UploadOut of these, Upon submit, the details shall be recorded in the database and the respective users shall be notified. Upon cancel, The input data shall be discarded.

CHAPTER VII

SYSTEM DESIGN

7.1 SYSTEM FLOW DIAGRAM

A system flow or activity diagram is a graphical representation of the flow of activities within a system or a process. It belongs to the Unified Modeling Language (UML) and is widely used in software engineering to model the dynamic aspects of a system. The primary purpose of an activity diagram is to depict the workflow or the sequence of activities, actions, and decisions that occur in a particular process.

In an activity diagram, activities are represented by rounded rectangles, and arrows show the flow of control between these activities. It helps to illustrate the order in which activities are executed, the conditions that govern their execution, and the parallel or alternative paths within the process. Decision nodes, represented by diamonds, indicate points where the flow of control can take different paths based on certain conditions.

Activity diagrams are valuable for understanding, documenting, and communicating the logic of a system's processes. They are particularly useful in business process modeling, software development, and system analysis. By visually representing the dynamic aspects of a system, stakeholders can gain insights into how different components or actors interact and collaborate to achieve specific goals.

In summary, a system flow or activity diagram is a powerful tool for capturing the dynamic behavior of a system, helping stakeholders visualize and analyze the sequence of activities and decision points within a process.

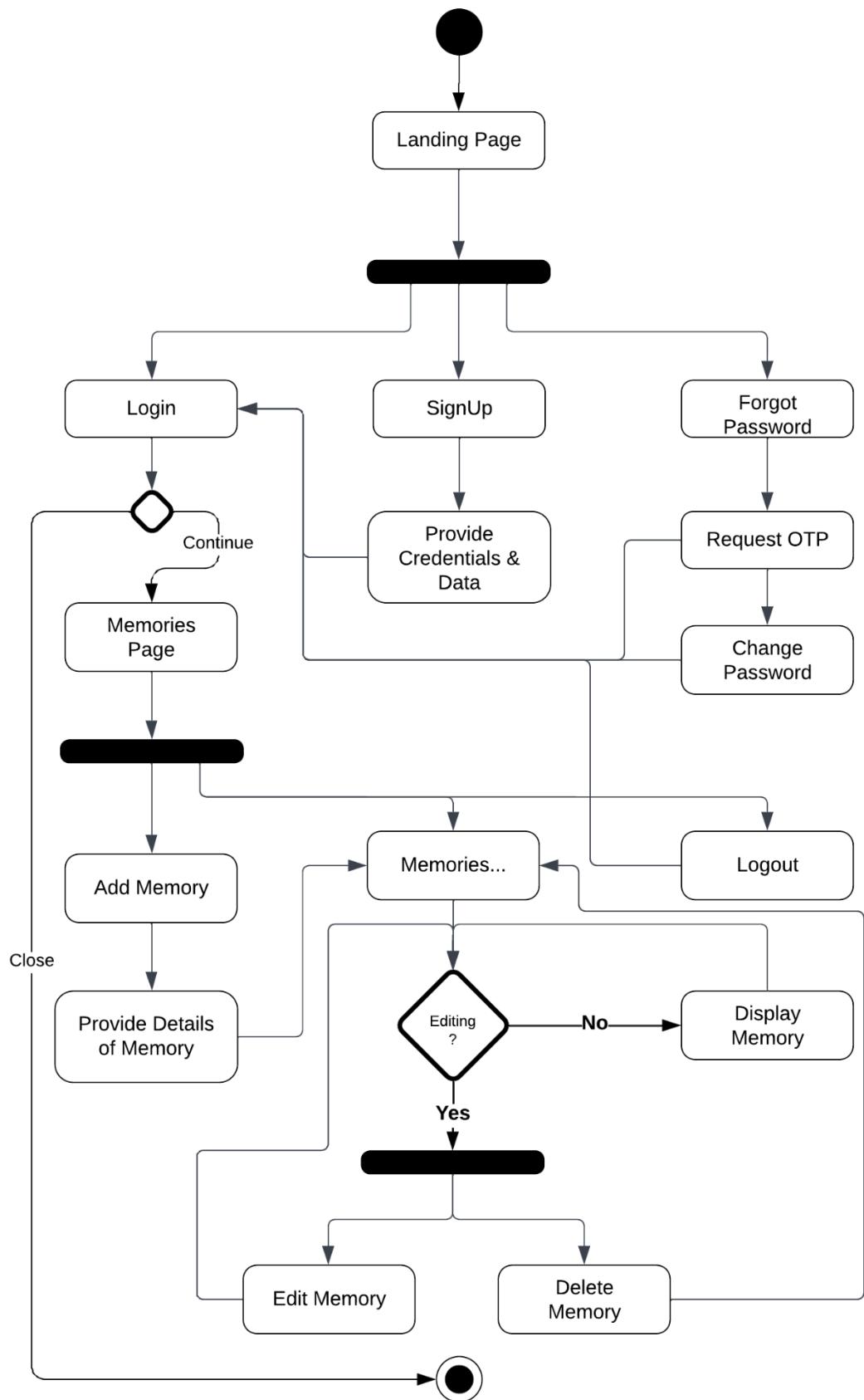


Fig.3 Flow diagram

CHAPTER VIII

SOURCE CODE

8.1 App.js

```
import {
  BrowserRouter as Router,
  Routes,
  Route,
  useNavigate
} from "react-router-dom";
import React from 'react';

import Landing from './pages/landing.page.jsx';
import Signup from './pages/authentication/signup.page.jsx';
import ForgotPassword from
'./pages/authentication/forgotPassword.page.jsx';
import ResetPassword from
'./pages/authentication/resetPassword.page.jsx';
import Memories from "./pages/memories.page.jsx";
import AddMemory from "./pages/addMemory.page.jsx";

function App() {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<Landing navigate={useNavigate}>/}>
        <Route path="/signup" element={<Signup
          navigate={useNavigate}>/}>
        <Route path="/forgot-password" element={<ForgotPassword
          navigate={useNavigate}>/}>
        <Route path="/reset-password" element={<ResetPassword
          navigate={useNavigate}>/}>

        <Route path="/memories" element={<Memories
          navigate={useNavigate}>/}>
        <Route path="/add-memory" element={<AddMemory
          navigate={useNavigate}>/}>
      </Routes>
    </Router>
  );
}

export default App;
```

8.2 Memories Page

```
import * as React from 'react';
import Button from '@mui/material/Button';
import CssBaseline from '@mui/material/CssBaseline';
import Grid from '@mui/material/Grid';
import Stack from '@mui/material/Stack';
import Box from '@mui/material/Box';
import Typography from '@mui/material/Typography';
import Container from '@mui/material/Container';
import { ThemeProvider, createTheme } from '@mui/material/styles';
import NavBar from '../components/navbar.component.jsx';
import BottomNavBar from
'../components/navbar_bottom.component.jsx';
import axios from 'axios';

import image from '../res/images/bg.jpg';
import MemoryCard from '../components/memoryCard.component.jsx';

const theme = createTheme({
  palette: {
    mode: 'dark',
    primary: {
      main: "rgba(255, 255, 255, 0.7)"
    },
    secondary: {
      main: "#E9C46A"
    }
  },
});

export default function Memories(props) {
  const [name, setName] = React.useState("");
  const [memories, setMemories] = React.useState([]);

  const loadData = async () => {
    await axios.get('http://localhost:2003/user/get-details',
    {withCredentials: true}).then((response) => {
      setName(response.data.name);
    });
    await axios.get('http://localhost:2003/memory/view-memory',
    {withCredentials: true})
    // .then((response) => response.json())
    .then((response) => {
      setMemories(response.data);
    })
  };
}
```

```

    });
}

React.useState(() => {
  loadData();
}, []);
return (
  <ThemeProvider theme={theme}>
    <CssBaseline />
    <NavBar></NavBar>
    <main>
      {/* Hero unit */}
      <Box
        component="div"
        sx={{
          position: 'fixed',
          width: '100%',
          height: '100%',
          backgroundImage: `url(${image})` ,
          backgroundSize: 'cover',
          backgroundRepeat: 'no-repeat',
          zIndex: -1000
        }}
      />
      <Container maxWidth="sm" sx={{pt: 12, pd: 8}}>
        <Typography
          component="h1"
          variant="h2"
          align="center"
          color="text.primary"
          gutterBottom
        >
          Welcome ! <br></br>{name}
        </Typography>
        <Typography variant="h5" align="center" color="text.secondary">
          paragraph>
          I am a passionate photographer who captures the beauty of life
          through the lens, turning moments into timeless art. My work is a visual
          storytelling that speaks volumes without words.
        </Typography>
        <Stack
          sx={{ pt: 4 }}
          direction="row"
          spacing={2}
          justifyContent="center"
        >

```

```

        <Button variant="contained" onClick={() => {window.location
= '/add-memory'}}>+ Add Memory</Button>
        <Button variant="outlined">Recent Memory</Button>
    </Stack>
</Container>
<Container sx={{ py: 8 }} maxWidth="xl">
    {/* End hero unit */}
    <Grid container spacing={4}>
        {memories.map((memory) => (
            <Grid item key={memory._id} xs={12} sm={6} md={4}>
                <MemoryCard
                    _id={memory._id}
                    description={memory.description}
                    title={memory.title}
                    memory_image={memory.memory_image}
                    isEditing={false}
                />
            </Grid>
        )));
    </Grid>
</Container>
    {/* </Box> */}
</main>
<BottomNavBar></BottomNavBar>
</ThemeProvider>
);
}

```

8.3 Login Page

```

import React from "react";
import { useNavigate } from "react-router-dom";

import '../res/css/bootstrap.min.css';
import '../res/css/materialdesignicons.min.css';
import '../res/css/style.min.css';

import bg from '../res/images/bg.jpg';

export default function Landing() {
    const [email, setEmail] = React.useState('');
    const [password, setPassword] = React.useState('');

    const navigate = useNavigate();

    async function handleChange(event) {

```

```

        if (event.target.id === 'email') {
            setEmail(event.target.value);
        } else if (event.target.id === 'userpassword') {
            setPassword(event.target.value);
        }
    }

    const validateEmail = (email) => {
        if(email.indexOf('@') < 1 || email.indexOf('.') < email.indexOf('@')
+ 1) {
            return false;
        } else {
            return true;
        }
    }

    async function submit(event) {
        if (validateEmail(email)) {
            event.preventDefault();
            const response = await fetch('http://localhost:2003/user/login', {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json'
                },
                body: JSON.stringify({
                    email: email,
                    password: password
                }),
                credentials: 'include'
            });
            if (response.status === 200) {
                navigate('/memories');
            } else {
                alert('Invalid email or password');
            }
        } else {
            alert('Invalid email');
        }
    }

    async function signup() {
        navigate('/signup');
    }

    return (
        <div className="vh-100 bg-account-pages" style={{{
backgroundImage: `url(${bg})`, backgroundSize: 'cover' }}>

```

```

<div className="container">
  <div className="row justify-content-center">
    <div className="col-md-8 col-lg-6">
      <div className="p-4">
        <div className="card overflow-hidden mt-2">
          <div className="text-center bg-primary position-relative">
            <div className="position-relative pt-4 py-5 mb-1">
              <h5 className="text-white">Welcome to Memory Vault!</h5>
              <p className="text-white fs-14 mb-0">Sign in to Continue...</p>
            </div>
          </div>
          <div className="card-body position-relative">
            <div className="p-4 mt-n5 bg-white card rounded pb-0">
              <form onSubmit={submit}>
                <div className="mb-3">
                  <label className="fs-14 mb-2" htmlFor="email">Email</label>
                  <input type="text" className="form-control" id="email" placeholder="Enter Email" onChange={handleChange}/>
                </div>
                <div className="mb-2">
                  <label className="fs-14 mb-2" htmlFor="userpassword">Password</label>
                  <input type="password" className="form-control" id="userpassword" placeholder="Enter Password" onChange={handleChange}/>
                </div>
                <div className="row">
                  <div className="col-sm-6"></div>
                  <div className="col-sm-6">
                    <div className="text-sm-end">
                      <a href="/forgot-password" className="text-muted fs-13"><i className="mdi mdi-lock-me-1" style={{color: "#ccc", font: "normal 14px/1 sans-serif; font-weight: normal; font-style: normal; font-variant: normal; font-size: 1em; line-height: 1; vertical-align: middle; width: 1em; height: 1em; display: inline-block; border: none; margin: 0; padding: 0;}}></i> Forgot password?</a>
                    </div>
                  </div>
                </div>
                <div className="mt-4">
                  <button className="btn btn-secondary w-100" type="submit">Log in</button>
                </div>
              </form>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

```

        </div>
    </div>
    <div className="mt-5 text-center text-white-50"
style={{display: "flex", justifyContent: "center"}}>
        <p style={{ color: 'white' }}>Don't have an account ?</p> &nbsp;&nbsp;
        <p className="fw-bold button" onClick={ signup } style={{ color: "white" }}> Register</p>
    </div>
    </div>
    </div>
    </div>
</div>
);
}

```

8.4 Memory Card Component

```

import * as React from 'react';
import Button from '@mui/material/Button';
import Card from '@mui/material/Card';
import CardActions from '@mui/material/CardActions';
import CardContent from '@mui/material/CardContent';
import CardMedia from '@mui/material/CardMedia';
import Typography from '@mui/material/Typography';
import axios from 'axios';
import TextField from '@mui/material/TextField';

const handleDelete = (id) => {
    axios.post('http://localhost:2003/memory/delete-memory', {id: id}, {withCredentials: true}).then(response => {
        window.location.reload();
    });
};

export default function MemoryCard(props) {
    const [isEditing, setIsEditing] = React.useState(props.isEditing);
    const [button, setButton] = React.useState({text: 'Edit'});
    const [content, setContent] = React.useState({
        title: null,
        description: null
    });
}

```

```

const handleChange = (event, parent) => {
  if (parent === 'title') {
    setContent({
      title: event.target.value
    });
  } else if (parent === 'description') {
    setContent({
      description: event.target.value
    });
  }

  if(content.title === null) {
    setContent({title: props.title});
  } else if (content.description === null) {
    setContent({description: props.description});
  }
}

const handleEdit = (event) => {
  if (!isEditing){
    setIsEditing(true);
    setButton({text: 'Done'});
  }
  else {
    if (content.title === null && content.description === null);
    else {
      axios.post('http://localhost:2003/memory/update-memory', {
        _id: props._id,
        title: content.title,
        description: content.description
      },
      {
        withCredentials: true
      })
      window.location.reload();
    }
    setIsEditing(false);
    setButton({text: 'Edit'});
  }
}

const description = (isEditing, description, title) => {
  console.log(isEditing);
  if (!isEditing) {
    return(

```

```

    <>
    <Typography gutterBottom variant="h5" component="h2">
      {title}
    </Typography>
    <Typography
      sx={{
        color: 'white'
      }}
      align='justify'
    >
      {description}
    </Typography>
    </>
  )
} else {
  return(
    <>
    <TextField
      id="outlined-static"
      label="Title"
      defaultValue={title}
      onChange={(event) => {handleChange(event, 'title');}}
      sx={{
        mb: 1,
        width: '100%'
      }}
    />
    <TextField
      id="outlined-multiline-static"
      label="Decription"
      multiline
      rows={9}
      defaultValue={description}
      onChange={(event) => {handleChange(event,
        'description');}}
      sx={{
        mb: 1,
        width: '100%'
      }}
    />
    </>
  )
}
}

return (
  <Card

```

```

        sx={{{
            height: '100%',
            display: 'flex',
            flexDirection: 'column'
        }}}
    >
    <CardMedia
        component="div"
        sx={{{
            pt: '56.25%',
        }}}
        image={props.memory_image}
    />
    <CardContent sx={{ flexGrow: 1 }}>
        {description(isEditing, props.description, props.title)}
    </CardContent>
    <CardActions>
        <Button size="small">View</Button>
        <Button size="small" onClick={() =>
handleEdit()}>{button.text}</Button>
        <Button size="small" onClick={() =>
handleDelete(props._id)}>Delete</Button>
    </CardActions>
</Card>
);
}

```

8.5 Server.js

```

import express, { json } from "express";
import cors from "cors";
import mongoose from "mongoose";
import dotenv from "dotenv";
import morgan from "morgan";
import cookieParser from "cookie-parser";

import userRouter from "./routes/user.route.js";
import logoutRouter from "./routes/logout.route.js";
import memoryRouter from "./routes/memory.route.js";

dotenv.config();

const app = express();
const port = process.env.PORT;

app.use(cors({

```

```

    credentials: true,
    origin: [process.env.CORS_ACCEPTED_ORIGIN_1]
  }));
app.use(json());
app.use(morgan("tiny"));
app.use(cookieParser());

mongoose.connect(
  process.env.MONGODB_URI,
  {
    useNewUrlParser: true,
    useUnifiedTopology: true
  }
);
mongoose.connection.once('open', () => {
  console.log("Connection Established Successfully");
});

app.listen(port, () => {
  console.log(`Server is active on port : ${port}`);
});

app.get('/', (req, res) => {
  res.send("Test");
});

app.use("/user", userRouter);
app.use("/logout", logoutRouter);
app.use("/memory", memoryRouter);

```

8.6 Memory Route

```

import express from "express";

import isLoggedIn from "../middleware/isLoggedIn.middleware.js";
import memory from "../models/memory.model.js";
import { upload } from "../modules/file-handler.module.js";

const router = express.Router();

/* SECTION - I */
router.get('/view-memory', isLoggedIn, async (req, res) => {
  try {
    const memory_detail = await memory.find({ email: req.user.email });
  }
});

```

```

        if(memory_detail) {
            res.json(memory_detail);
        } else {
            res.status(400).json({ error: "No memory found" });
        }
    } catch (error) {
        res.status(400).json({ error });
    }
});
/* END OF SECTION - I */

/* SECTION - II */
router.post('/add-memory', isLoggedIn, async (req, res) => {
    try{
        if (req.file !== undefined) {
            req.body.memory_image = req.file.filename;
        }
        // console.log(req.user.email);
        const newMemory = await memory.create({
            email: req.user.email,
            description: req.body.description,
            memory_image: req.body.memory_image,
            location: req.body.location,
            title: req.body.title
        });
        newMemory.save().then(() => console.log("saved"));
        console.log(newMemory);
        res.status(200).json("Memory added");
    } catch (error){
        res.status(400).json({ error });
    }
});

router.post('/delete-memory', async (req, res) => {
    await memory.deleteOne({ _id: req.body.id }).then(() => {
        res.status(200).json("Memory deleted");
    });
});

router.post('/update-memory', async (req, res) => {
    await memory.findOneAndUpdate({ _id: req.body._id }, {
        title: req.body.title,
        description: req.body.description
    }).then(() => {
        res.status(200).json("Memory deleted");
    });
});

```

```
export default router;
```

8.7 Memory Model

```
import mongoose from "mongoose";

const Schema = mongoose.Schema({
  email: {
    type: String,
    required: true,
  },
  description: {
    type: String,
    required: true
  },
  title: {
    type: String,
    required: true
  },
  memory_image: {
    type: String,
    required: true
  },
  location: {
    type: String
  },
  time: {
    type: Date,
    default: Date.now()
  },
  {
    timestamps: false
  });

```

```
const memory = mongoose.model('memory', Schema);
```

```
export default memory;
```

CHAPTER IX

EXECUTION

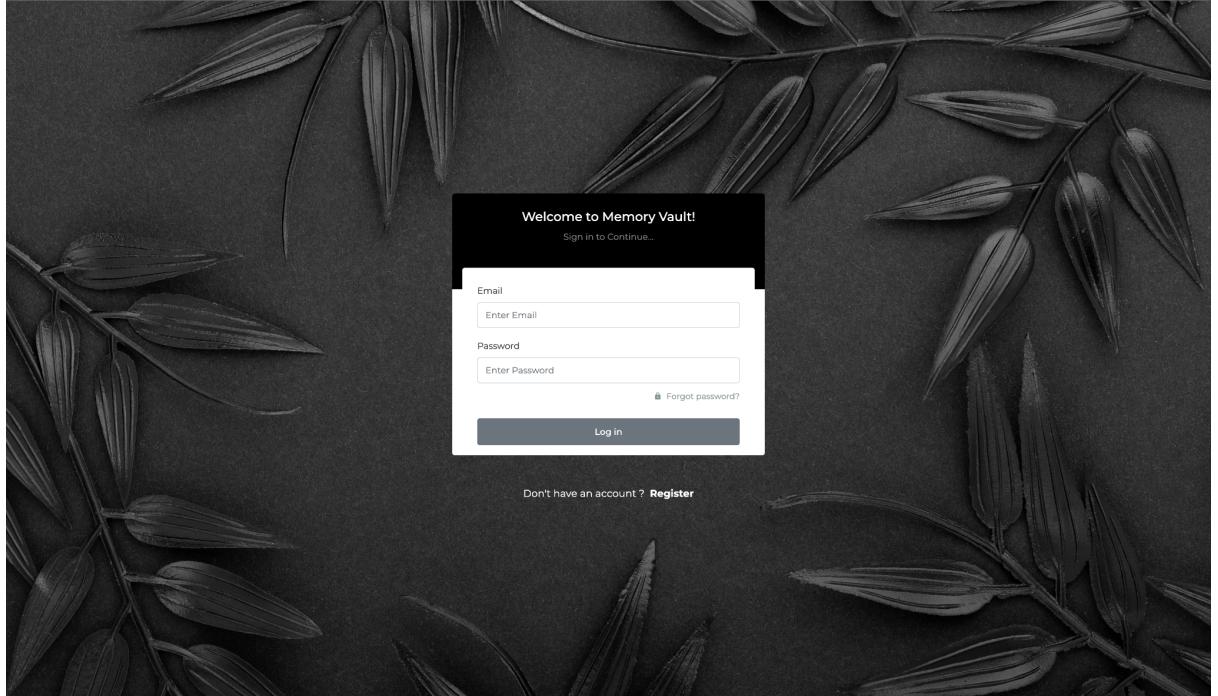


Fig.4 Login - Raw

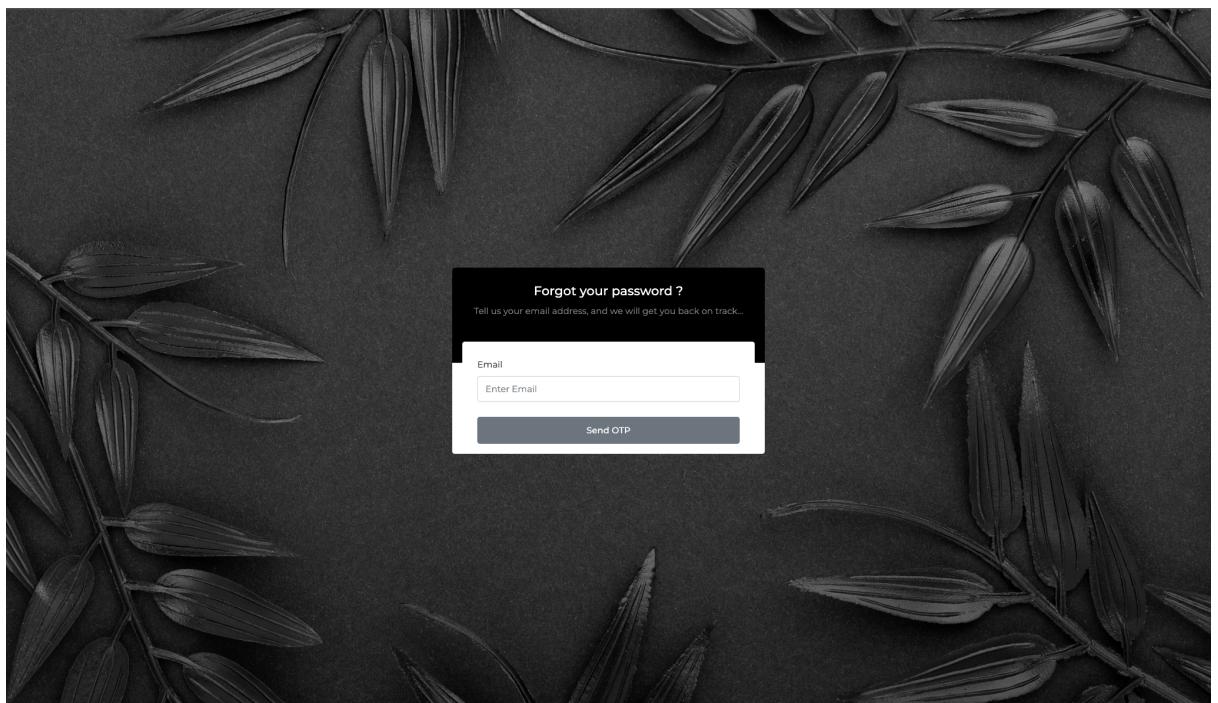


Fig.5 Forgot Password

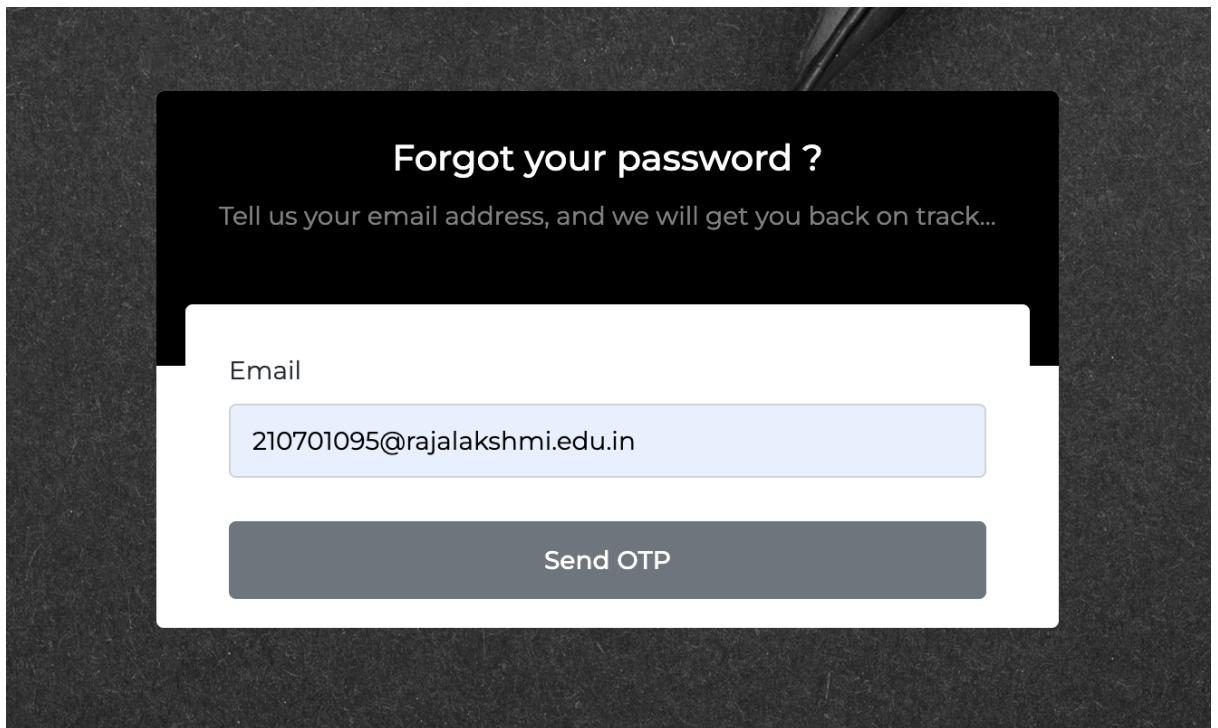


Fig.6 Forgot Password - With details

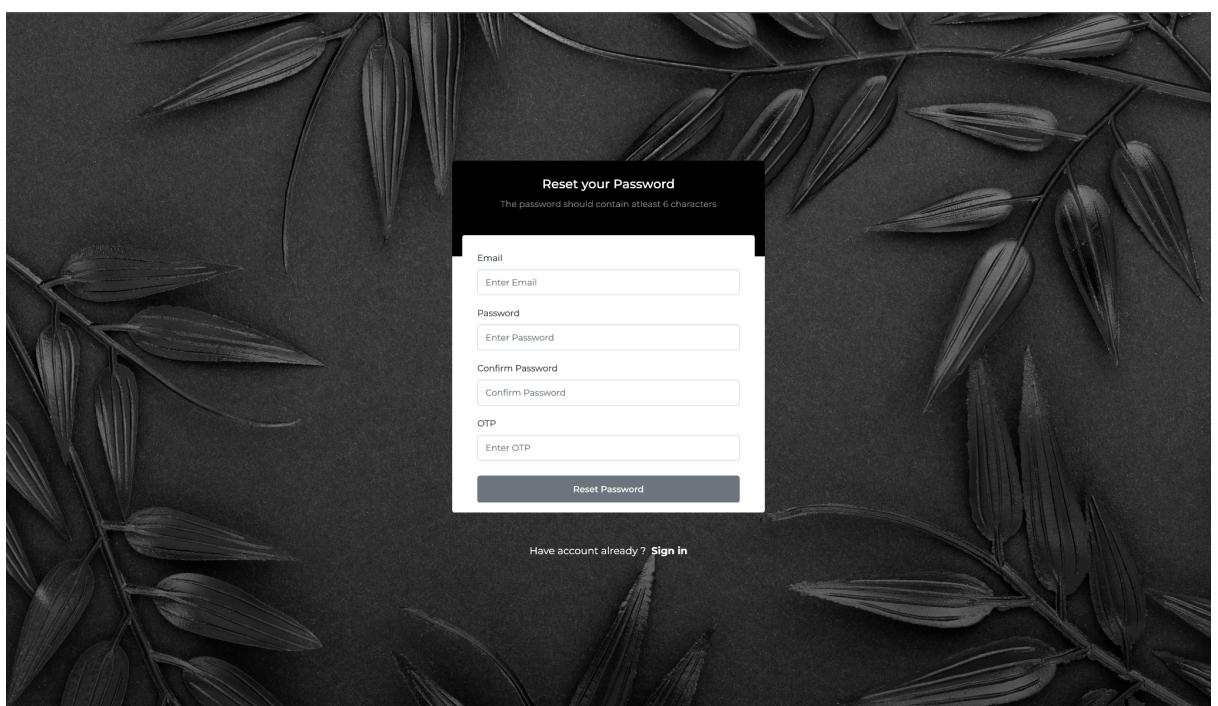


Fig.7 Reset Password

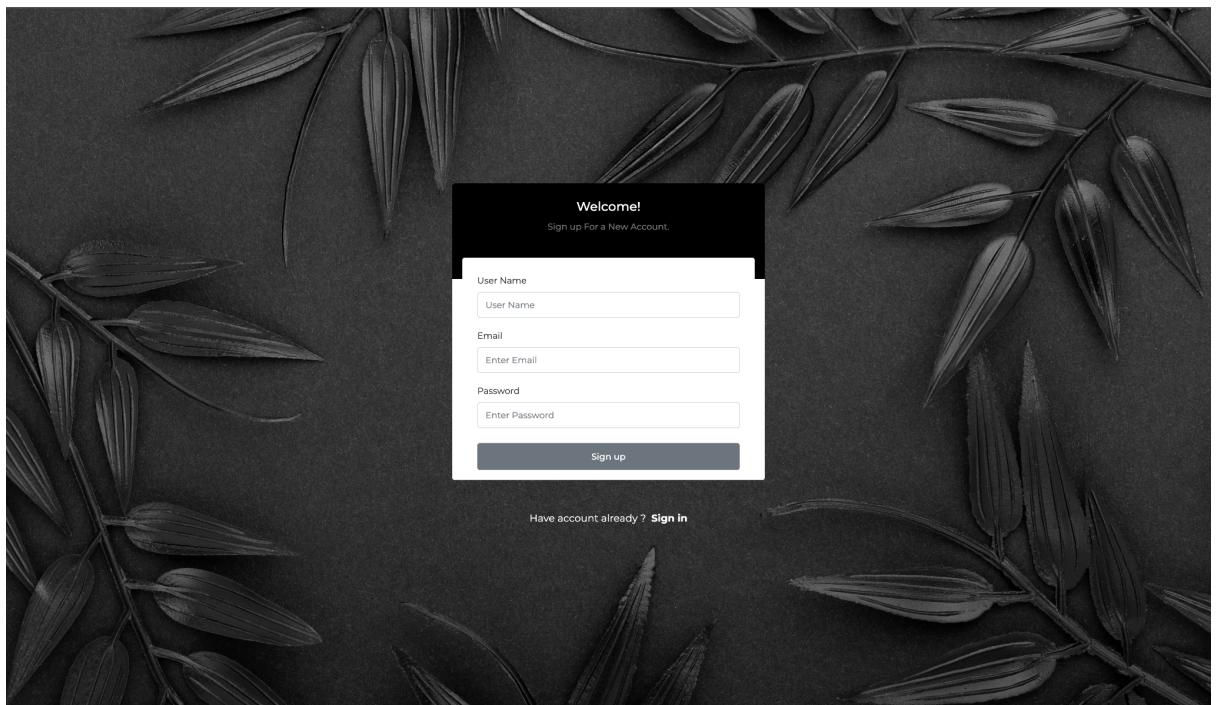


Fig.7 Reset Password

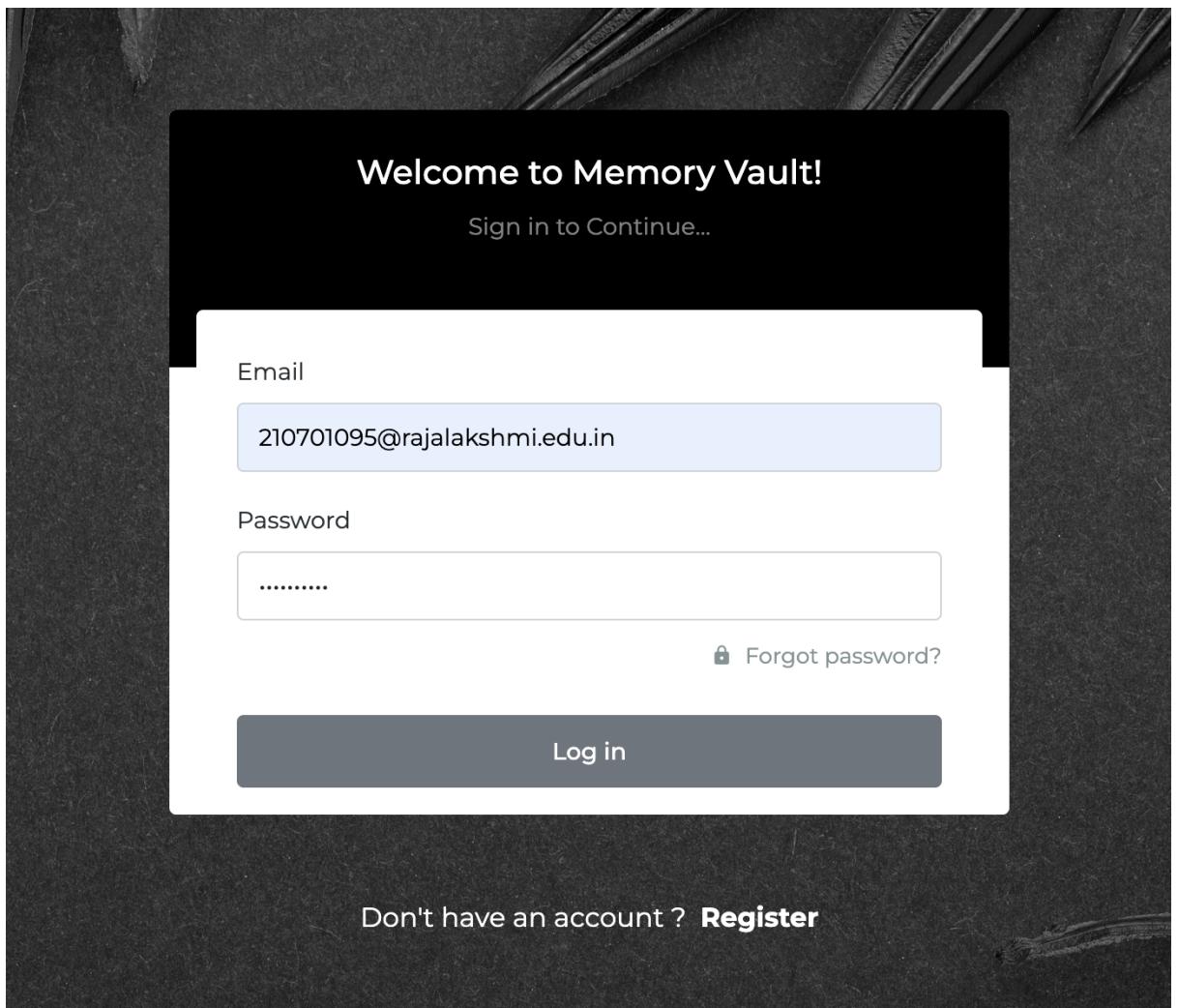


Fig.8 Login - With Details

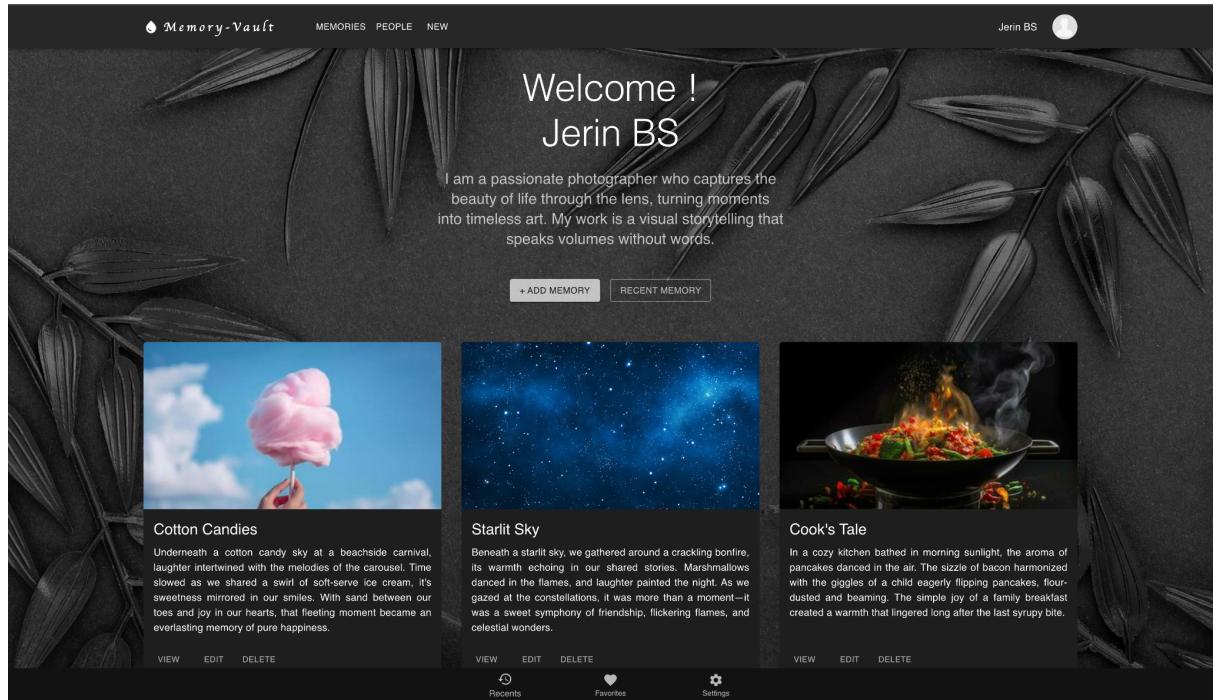


Fig.9 Memories

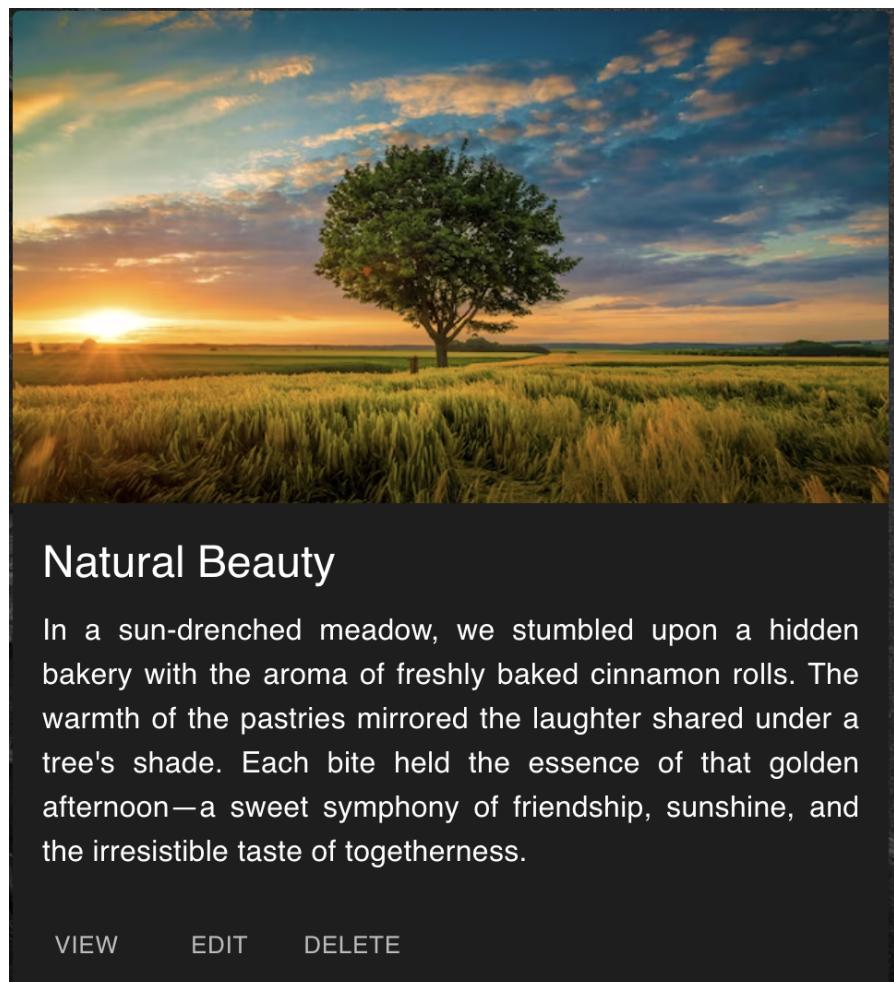


Fig.10 Memory

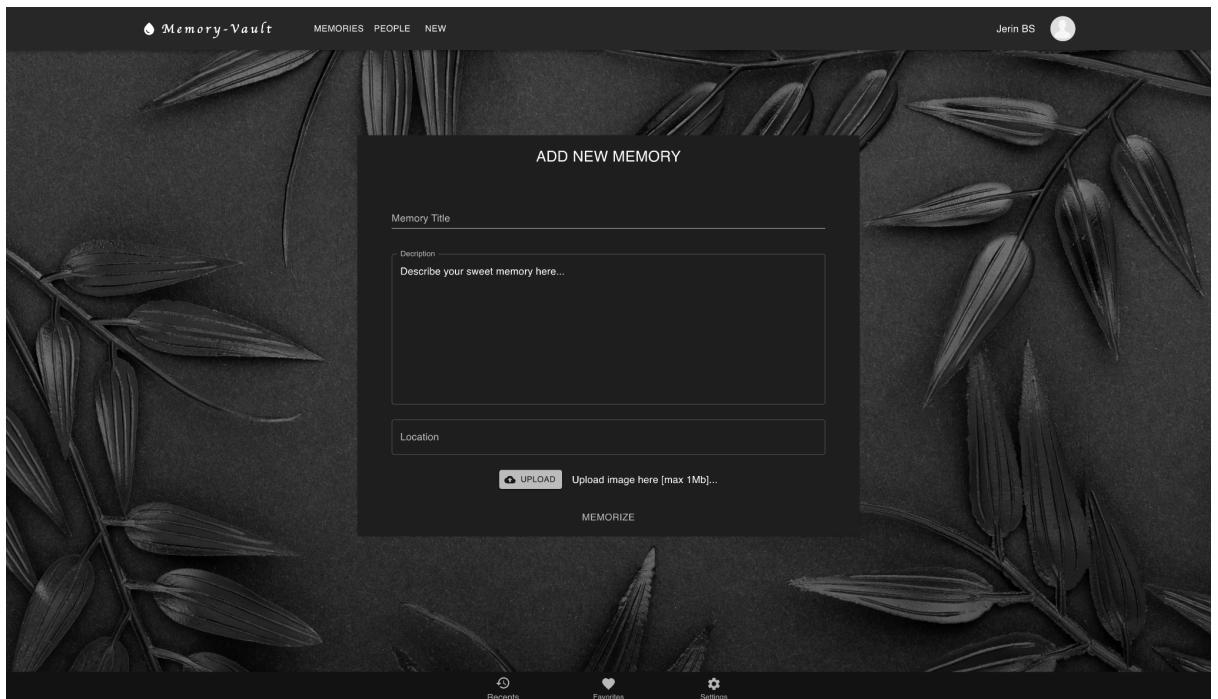


Fig.11 Add Memory

ADD NEW MEMORY

Memory Title
Willow Bed

Description
Beneath a willow tree's gentle shade, the aroma of freshly baked cookies mingled with the warmth of a shared blanket. As sunlight filtered through leaves, we savored homemade treats, filling the air with laughter and the taste of chocolate chips. In that serene haven, bonds deepened, etching a sweet chapter in our shared history.

Location
Garden

UPLOAD Weeping-Willow-1024x574.webp

MEMORIZZE

Fig.12 Add Memory - With Details

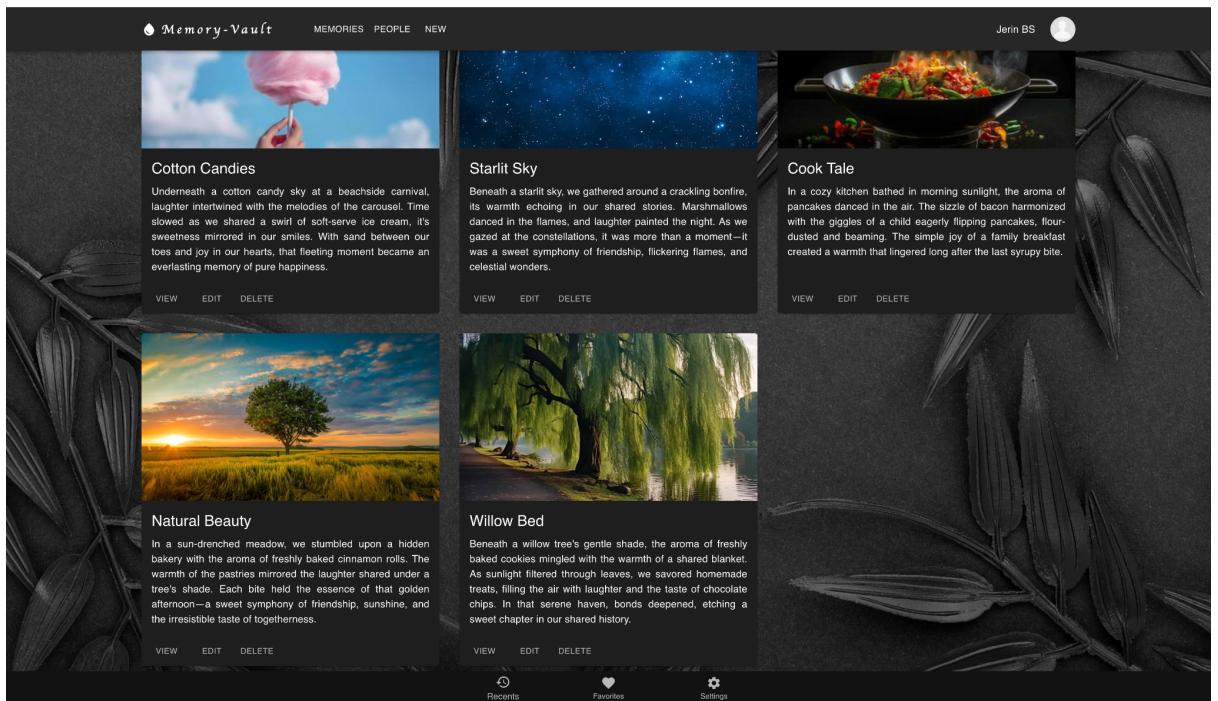


Fig.13 Memories - After Adding

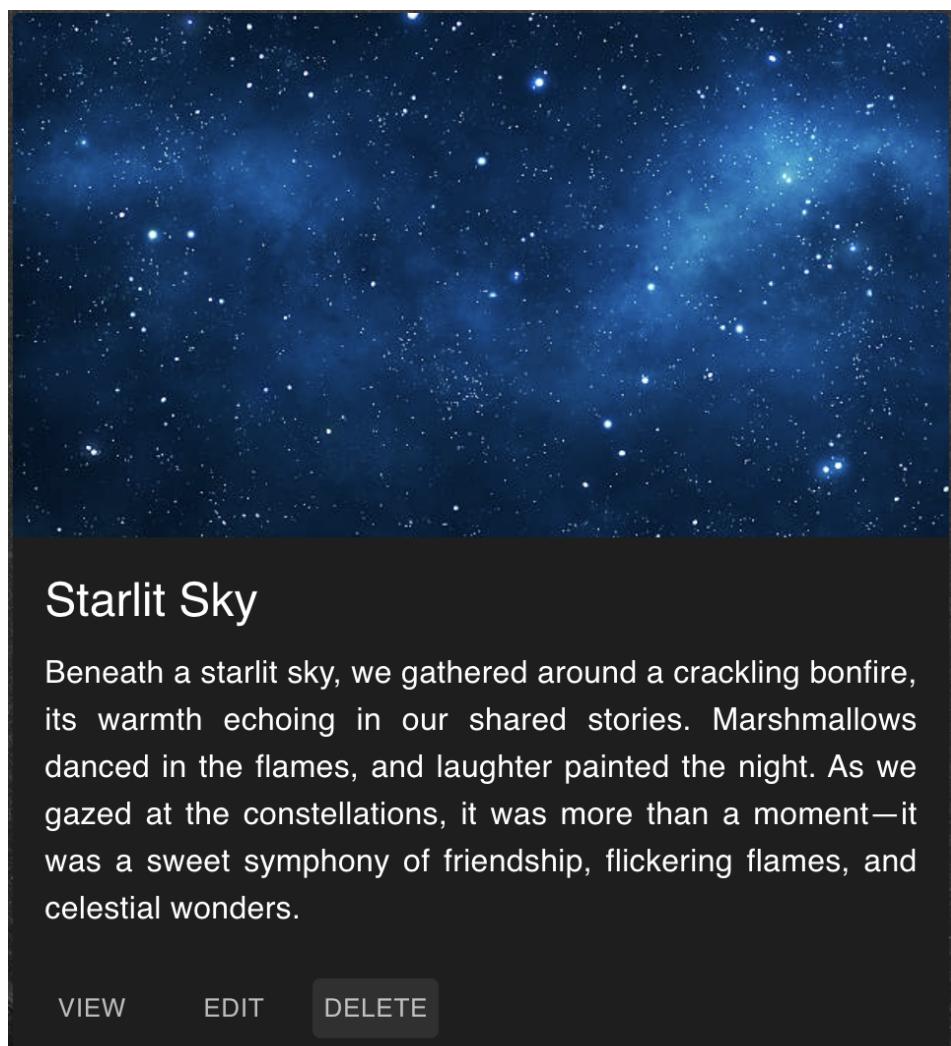


Fig.14 Memory - To Be Deleted

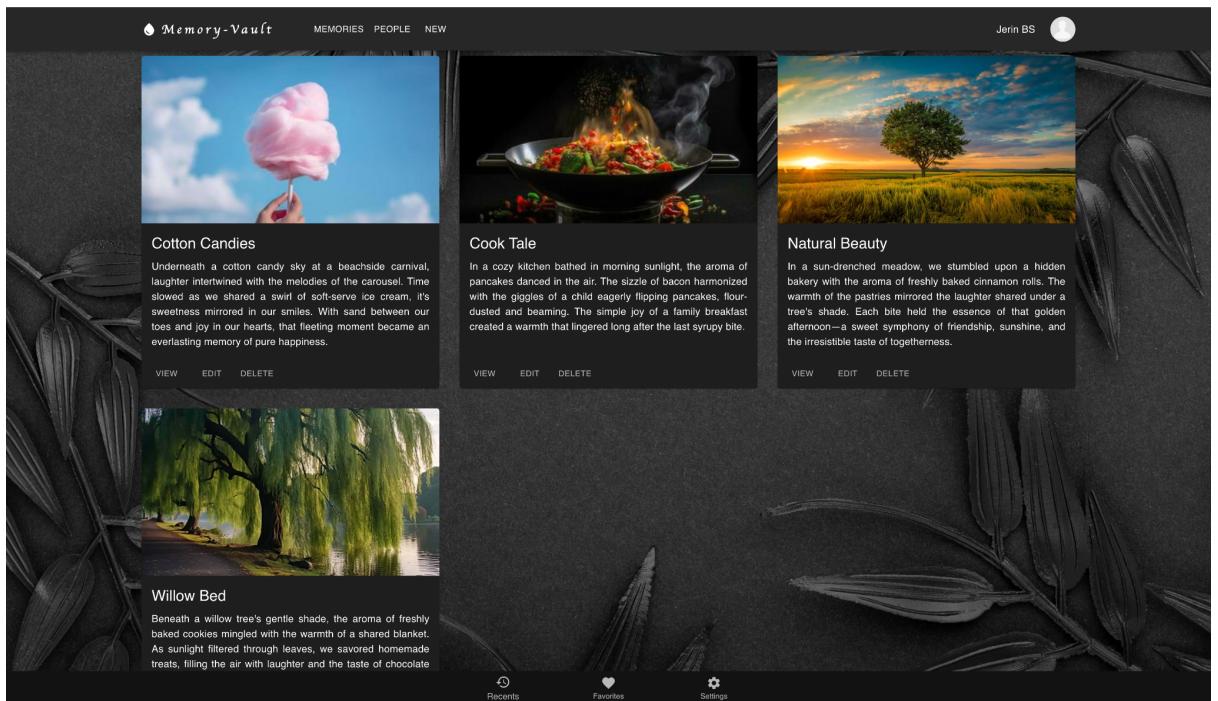


Fig.15 Memories - After Deleting



Fig.16 Editing A Memory

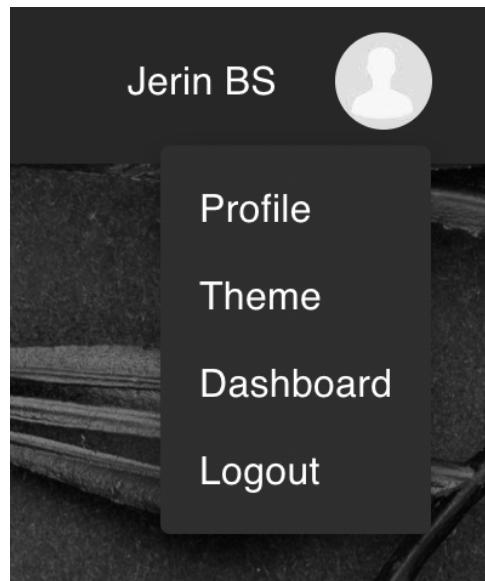


Fig.17 User Settings

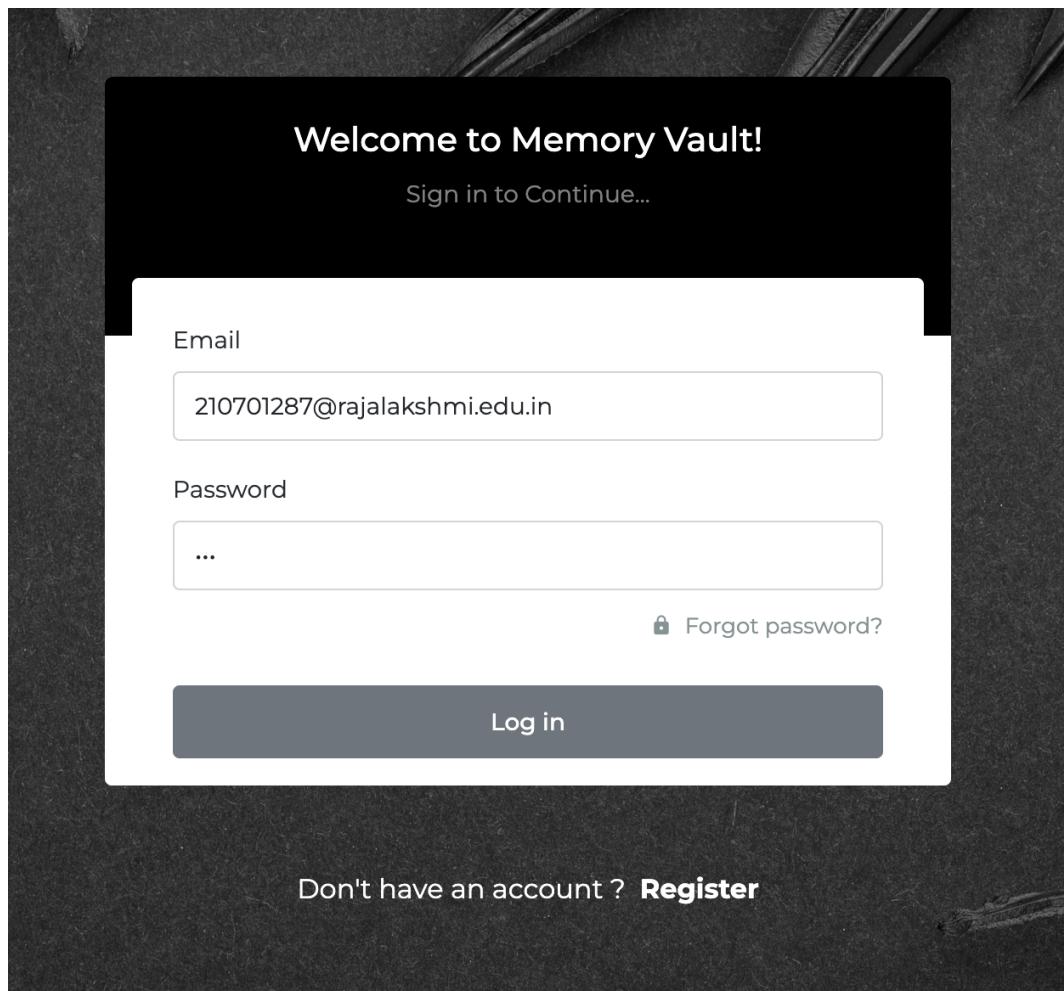


Fig.18 Login - Another Account

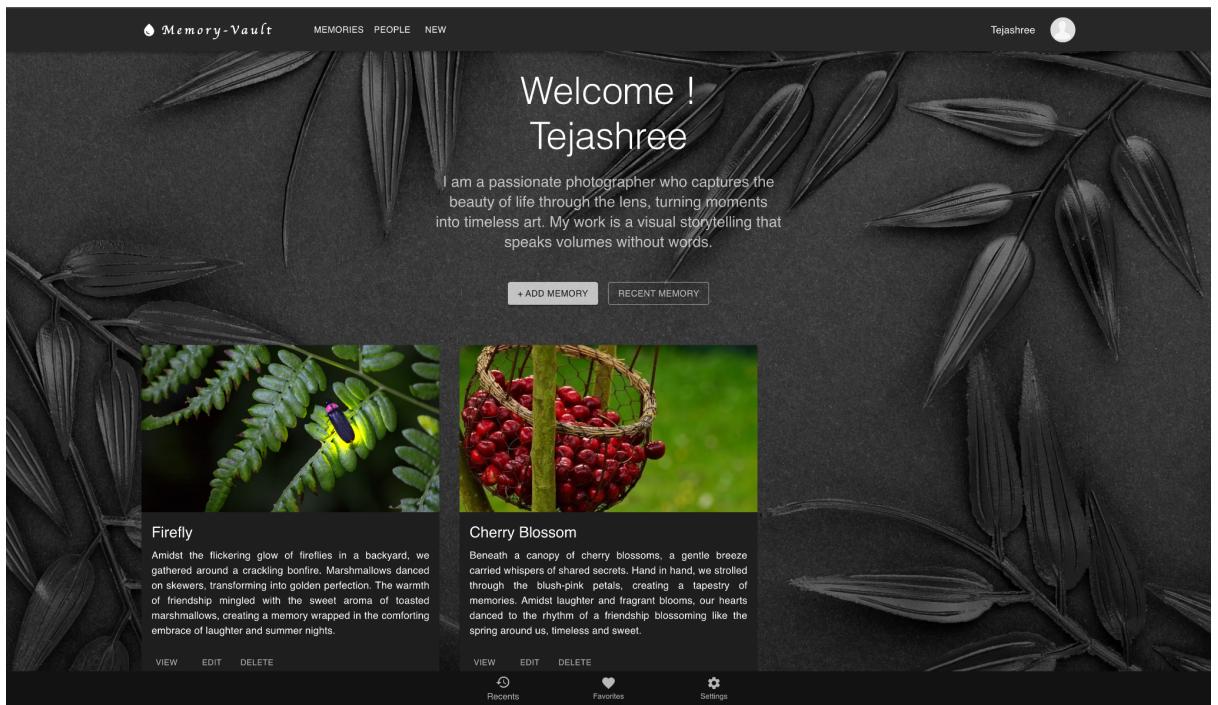


Fig.19 Memories - Another User

CHAPTER X

CONCLUSION & FUTURE ENHANCEMENTS

In conclusion, this project endeavors to redefine the digital memory management landscape by providing users with a secure, interactive, and user-friendly platform. By seamlessly integrating text, images, and location data, the application empowers users to craft a nuanced and holistic narrative of their lives. The commitment to data security, intuitive navigation, and collaborative features ensures a comprehensive and enriching experience.

Looking ahead, future enhancements could explore the integration of advanced artificial intelligence algorithms for personalized content recommendations, sentiment analysis of memories, and an expanded range of multimedia elements. Additionally, incorporating augmented reality features for an immersive memory exploration experience could further elevate the user engagement. Continuous refinement of the collaborative elements, such as real-time co-authoring of memories and expanded social sharing capabilities, would contribute to creating a truly dynamic and interconnected digital memory ecosystem.

CHAPTER XI

REFERENCES

1. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
2. <https://react.dev/learn>
3. <https://mui.com/material-ui/getting-started/overview/>
4. <https://fontawesome.com/icons>
5. <https://stackoverflow.com/questions/>
6. <https://www.geeksforgeeks.org/>
7. <https://www.bezkoder.com/>
8. https://www.w3schools.com/html/html_css.asp
9. <https://www.npmjs.com/package/>
10. <https://mongoosejs.com/docs/>