A Project Report on

TREASURE OF MEMORIES

Submitted by

S Sarfraj – R170317

L Umar Ahmad – R170930

P Jaya Sreenivas – R171005

Submitted to

IIIT RK VALLEY
Idupulapaya, Vempalli, YSR Kadapa
Andhra Pradesh, India PIN 516330

Under the guidance of

M HimaBindu

Assistant Professor

As a part of

Partial fulfilment of the degree of Bachelor of Technology in Computer Science and Engineering

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

R.K.Valley, Kadapa (Dist), Andhra Pradesh,516330

CERTIFICATE

This is certify that the project entitled “**TREASURE OF MEMORIES”** submitted by**S.Sarfraj** (**R170317**), **L.Umar Ahmad** (**R170930**), **P.Jaya Sreenivas** (**R171005**), under our guidance and supervision for the partial fulfilment for degree Bachelor ofTechnology inComputer Science and Engineering – 3 during the academic semester – II 2021-2022 at RGUKT, RK VALLEY**.** To the best of our knowledge, the report has not been submitted previously in part or in full to this or any other University or Institution for the award of anydegree or diploma

Project Internal Guide	**Head of the Department**
M HimaBindu,	P Harinadha,
Assistant Professor	Assistant Professor,
Computer Science and Engineering	Computer Science and Engineering,
RK Valley, RGUKT.	RK Valley, RGUKT.

Submitted for the practical examination held on ....

Internal Examiner **External Examiner**

Acknowledgement

We would like to express our sincere gratitude to **Ms. M HimaBindu** Mam, our projectinternal guide for valuable suggestions and keen interest throughout the progress of my course of research.We are grateful to **Mr**. **P Harinadha** sir, HOD CSE, for providing excellent computing facilities and a congenial atmosphere for progressing with our project.

At the outset, we would like to thank **Rajiv Gandhi University of Knowledge Technologies** for providing all the necessary resources for the successful completion of our course work.

Index

Abstract

It is a web application which is used for storing the memories which we have cherished the moments in our life which are close to our heart. It’s a type of a vlog/memory diary in which you can store a group of photos and you can write down the moments of some trips/tours/events.This web application is opensource and any user can use it and each have their own personal account which is for themselves. This consists of three fields namely profile, memories, new memories.

Introduction

This document has the requirements of Treasure of Memories. Treasure of Memories is used to save the pictures that are taken when you visit to the certain places in your tour.

In this the stored each memories of an user will be display in separate vlogs

1.1: Purpose

The purpose of this document is to create your tout vlog where you can place the set images that are taken on that tour.

1.2: Intended Audience:

The intended audience will be the user, were he can upload the photos of the tour and access the pictures of his tour.

Users

Product Vision:

Treasure of Memories product is design to save the photos of the trips that were done by the user. The Main intention of this product is to Place set of photos in single place were don’t have fear of deleting of photos that were taken in trip.

Technologies:

➢ HTML

➢CSS

➢Java script

➢Node JS

➢Express JS

➢ MongoDB

Node JS :

Node JS is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.

Mongo DB :

Mongo DB is a document-oriented NoSQL database used for high volume data storage. Instead of using tables and rows as in the traditional relational databases, MongoDB makes use of collections and documents. Documents consist of key-value pairs which are the basic unit of data in Mongo DB. Collections contain sets of documents and function which is the equivalent of relational database tables.

Collections → Table

Documents → Rows

Express JS :

Express is a node js web application framework that provides broad features for building web and mobile applications. It is used to build a single page, multipage, and hybrid web application.

It's a layer built on the top of the Node js that helps manage servers and routes

Express was created to make APIs and web applications with ease,

It saves a lot of coding time almost by half and still makes web and mobile applications are efficient.

Another reason for using express is that it is written in JavaScript as java script is an easy language even if you don't have a previous knowledge of any language. Express lets so many new developers enter the field of web development.

The reason behind creating an express framework for Node JS is

➢Time Efficient

➢Fast

➢Economical

➢Easy to learn

➢Asynchronous

Software Configurations

➢Node.js v16.17.0

➢Ubuntu 18.04 LTS

Design

Modules

- **Login**
- **Registration**
- **Profile**
- **New Memories**
- **Memories**

Login & Registration:

In this module if the user is new user, then he has to create an account by giving the user credentials according to the in registration form.

If the user has an already account,then user have to login with user Email id and password

Profile:

In this module the user details will be displayed

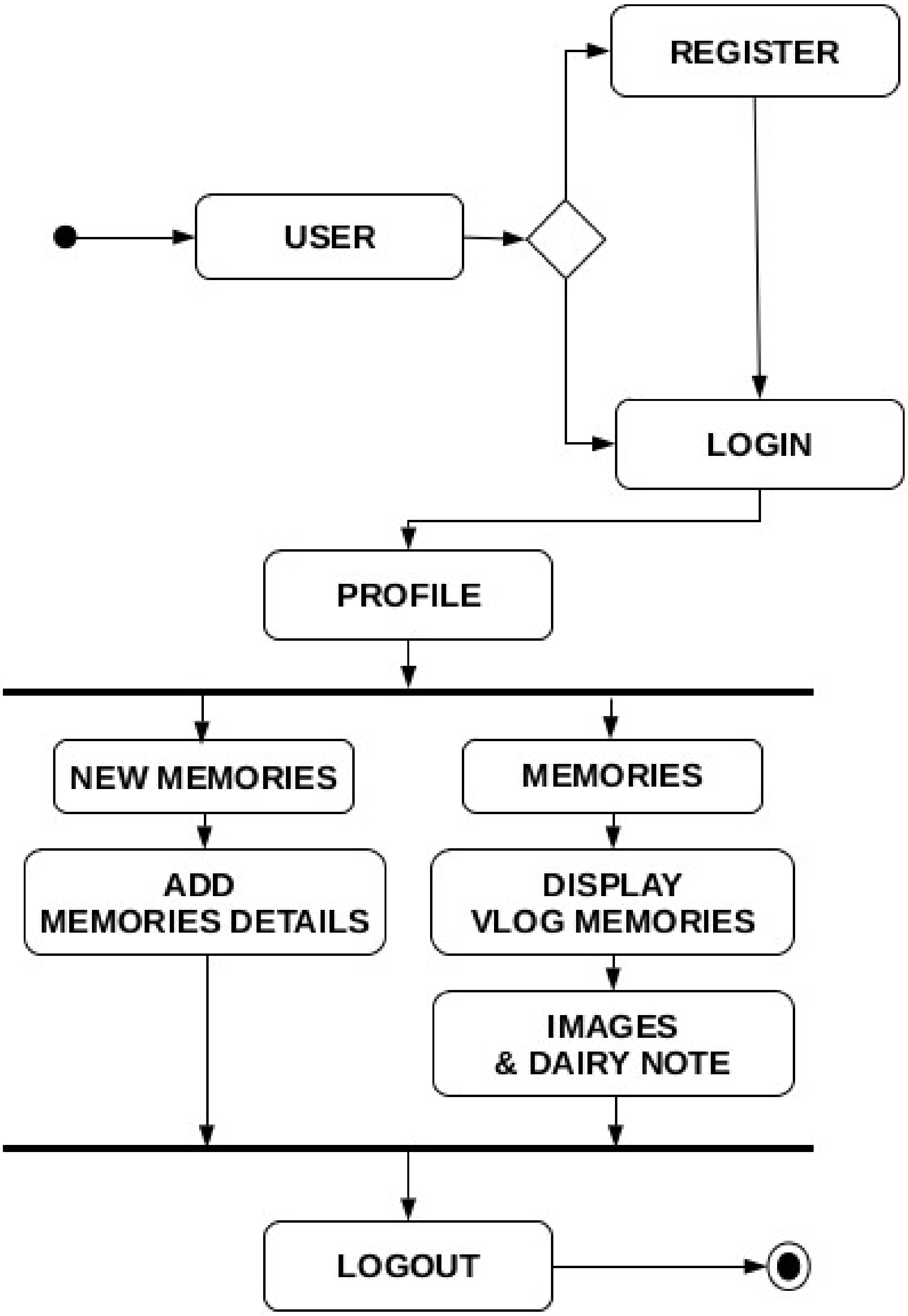
New Memories:

In this module the user can create new memories by giving the Title, Date,Images, and short note diary etc.

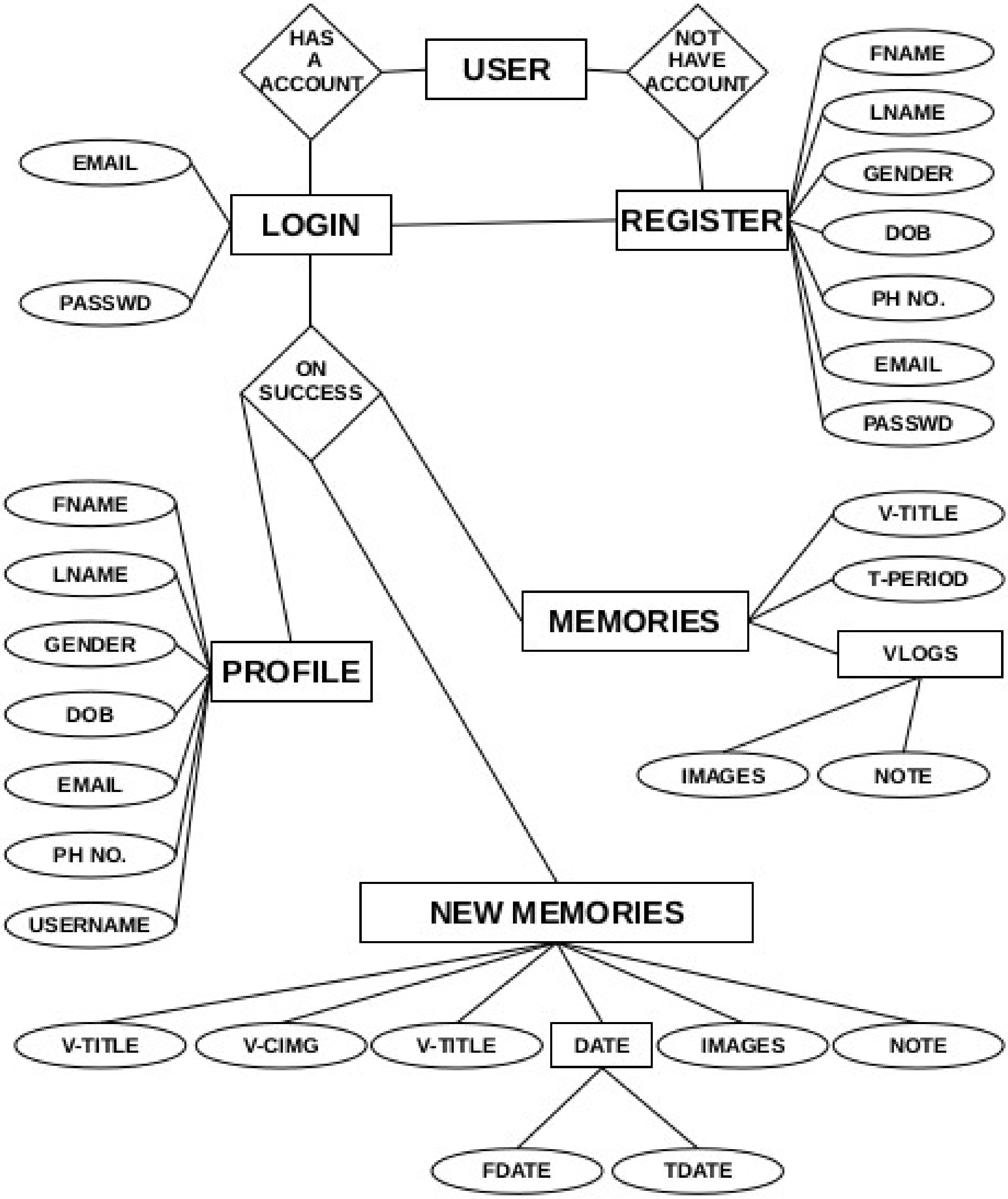
Memories:

In this module all of the user memories which he/she created will be displayed.

Activity Diagram :



ER Diagram :


HAS A ACCOUNT
USER
NOT HAVE ACCOUNT
FNAME
LNAME
GENDER
DOB
PH NO.
EMAIL
PASSWD
EMAIL
PASSWD
LOGIN
REGISTER
ON SUCCESS
FNAME
LNAME
GENDER
DOB
EMAIL
PH NO.
USERNAME
PROFILE
MEMORIES
V-TITLE
T-PERIOD
VLOGS
IMAGES
NOTE
NEW MEMORIES
V-TITLE
V-CIMG
V-TITLE
DATE
IMAGES
NOTE
FDATE
TDATE

Coding

Index.js:

```
import express, { response } from 'express'
import bodyParser from 'body-parser'
import mongoose from 'mongoose'
import { MongoClient } from 'mongodb';
import memories_colls from './models/memories_colls.js'
import users from './models/users.js'
import fs from 'fs-extra'
import { ObjectId } from 'mongoose';
import imageToBase64 from 'image-to-base64';
import path from 'path';
import multer from 'multer';
import multiparty from 'multiparty'
import fileupload from 'express-fileupload'
import cors from 'cors'

let count = 0;
var upload = multer({limits: {fileSize: 1064960
},dest:'./uploads/'}).single('picture');
const uploadDirectory=path.join(process.cwd(), 'public/HTML/uploads')
console.log(uploadDirectory)

const app = express();
app.use(cors())

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

app.use(bodyParser.urlencoded({
    extended:false
}));
app.use(express.json());
app.use(bodyParser.json());

const password = 'UseSarfrajData120'
const CONNECTION_URL = `mongodb+srv://Sarfraj:$
{password}@cluster0.rse9f.mongodb.net/?retryWrites=true&w=majority`
const PORT = process.env.PORT || 5000;


mongoose.connect(CONNECTION_URL, { useNewURLParser: true,
useUnifiedTopology: true })
    .then(() => app.listen(PORT, (req, res) => console.log(`Listening on port
${PORT}`)))
    .catch((error) => console.log(error.message) )


app.use('/', express.static('public/HTML'))


let userId = '';
var loginId = '';
app.post("/login", async (req, res) => {
  try {
    const { email, passwd } = req.body;
    console.log(email, passwd);
    MongoClient.connect(CONNECTION_URL, function(err, db) {
            if (err) throw err;
            var dbo = db.db("test");
            dbo.collection("users").findOne({ email }, function(err, user) {
              if (err) throw err;
              loginId = user._id.toString();
              console.log(loginId);


              if(user.password === passwd) {
                      return res.redirect('profile.html');
              } else {
                      return res.status(400).send("User not found");
              }
            });
    });
  } catch (err) {
            console.log(err);
            res.status(500).send("Something went wrong");
  }
  });


app.post('/sign_up', async (req, res) => {
    var first_name = req.body.fname;
  var last_name = req.body.lname;
  var dob = req.body.dob;
  var mobile = req.body.mobile;
  var gender = req.body.gender;
  var email =req.body.email;
  var pass = req.body.password;

  console.log(first_name, last_name);

    var data = {
        "first_name": first_name,
        "last_name": last_name,
        "dob": dob,
        "mobile": mobile,
        "gender": gender,
        "email":email,
        "password":pass,
        "memories":[[]]
  }
  const name = first_name + last_name;
    MongoClient.connect(CONNECTION_URL, function(err, db) {
        if (err) throw err;
        var dbo = db.db("test");
        dbo.collection("users").insertOne(data, function(err, res) {
                userId = res.insertedId.toString();

                console.log(userId);
          if (err) throw err;
          console.log("1 document inserted");
          db.close();
        });
  });
  res.redirect('index.html');
} )

app.post('/newMemory', (req, res) => {

  let form=new multiparty.Form({uploadDir:uploadDirectory})
```

```
form.parse(req,async (error,fields,files)=>{
if (error){
    res.send('error');
             return
}
   console.log(fields)
   console.log(files)
const fileName=files.vlog_cover[0].path
   const img1 = files.img1[0].path;
   const img2 = files.img2[0].path;
   const img3 = files.img3[0].path;
   const img4 = files.img4[0].path;
   const img5 = files.img5[0].path;
   const img6 = files.img6[0].path;

   var vlog_title = fields.vlog_title[0];
   var date_from = fields.date_from[0];
   var date_to = fields.date_to[0]
   var vlog_cover = fileName
   var imgs =0;
   var note = fields.note[0];
   count++;

   var data = {
             id: userId ,
             vlog_title: vlog_title,
             date_from: date_from,
             date_to: date_to,
             vlog_cover: vlog_cover,
             img1: img1,
             img2: img2,
             img3: img3,
             img4: img4,
             img5: img5,
             img6: img6,
             note: note,
             count: count
   }
   MongoClient.connect(CONNECTION_URL, function(err, db) {
```

```
                    if (err) throw err;
                    var dbo = db.db("test");
                    dbo.collection("memories_colls").insertOne(data, function(err,
res) {
                      if (err) throw err;
                      console.log("1 document inserted");
                    });
          });
          res.redirect('newMemories.html');
  })
});

let all_posts = [];
app.get('/memories', async (req, res) => {
        try {
            all_posts = await memories_colls.find();
            console.log(all_posts)

            res.status(200).json(all_posts);
        } catch (error) {
            res.status(404).json({ message: error.message })
        }
})
let all_users = []
app.get('/users', async (req, res) => {
  try {
          all_users = await users.findOne({id: loginId});
          res.status(200).json(all_users);
          req.body.first_name = all_users.first_name;
          console.log(all_users);
  } catch (error) {
          res.status(404).json({ message: error.message });
  }
});

app.get('/vlog/:id',  (req, res) => {
    var id = req.params.id;
  const images=[]
  memories_colls.findById(id, (err, result) => {
          if(err) {
```

```
                    console.log(err);
            } else {
                    console.log( "result:", result)
                    images.push(result.img1)
                    images.push(result.img2)
                    images.push(result.img3)
                    images.push(result.img4)
                    images.push(result.img5)
                    images.push(result.img6)
                    const note=result.note
                    res.render("eachMemory.ejs",{images,note})
            }
     })
    })
```

Testing

Here we performed two types of testing to the software for finding bugs

1.Functional Testing:

we tested main features like testing each and every module like login , signup, Profile, Memories, New Memories.

Integration Testing:

Here, the data flow is tested .For example ,if we take login moduleby entering valid credentials it redirects to the respected users Dashboard .

System Testing:

Here, the end to end Testing is done on application from entering credentials, navigating to the all modules such as Profile of the User etc. and at last to the logout page.

2.Non-Functional Testing:

Here we tested the Non-functional features like Compatibility, Performance

Compatibility Testing:

Here We tested this software on Various Operating System such , windows etc…

Performance Testing

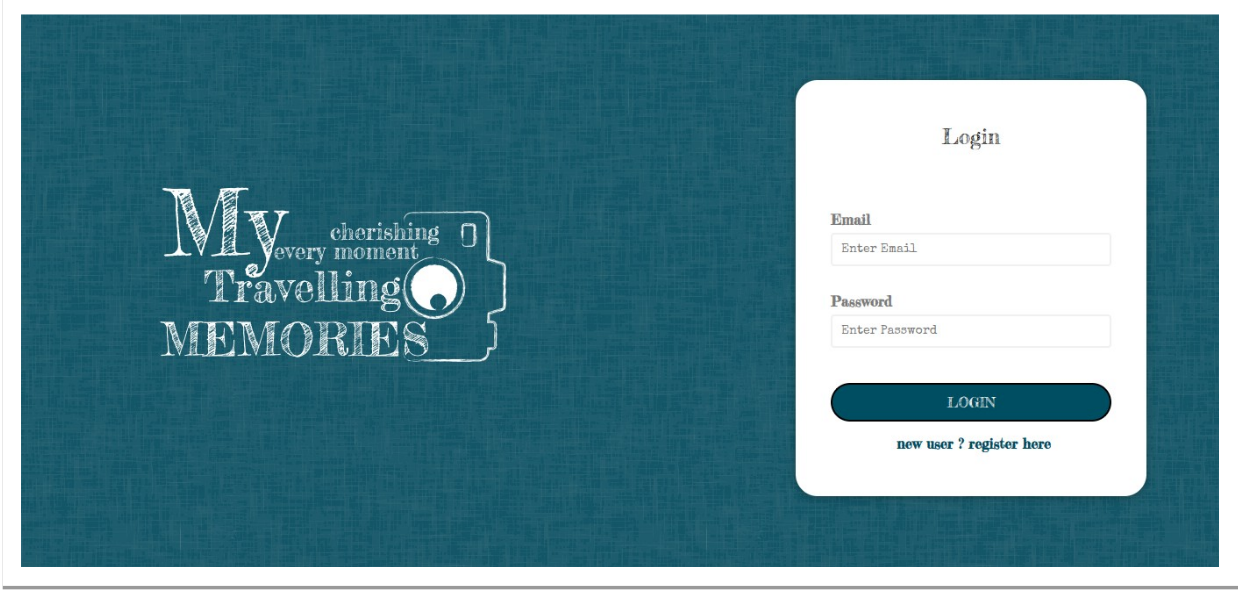
Here we tested the speed, efficiency. The software is given accurate results when the user enters the data.

Future Improvements

- Edit option to the profile
- Adding favourites option for the memories
- Increasing the limit of selecting number of images
- Modifying and deletion of the images of particular memories
- Reminding of memories of an user through the email on the same day every year

Module Snippets

Login Page:



Login

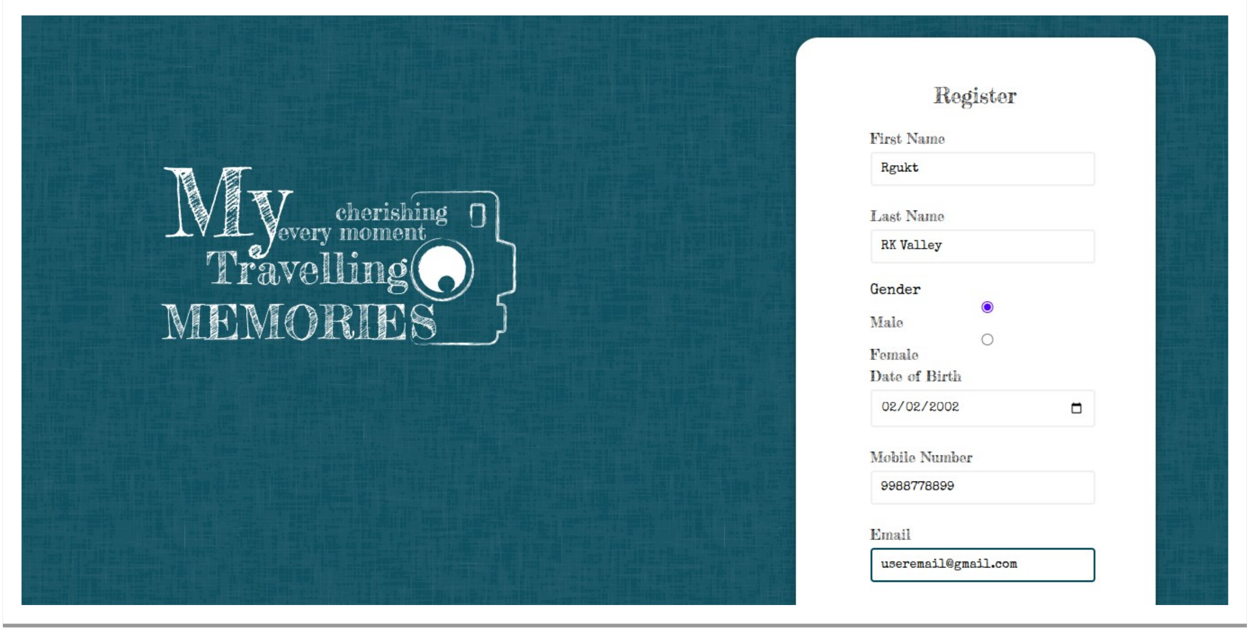
Email

Enter Email

Password

Enter Password

Register Page:



Register

First Name

Rgukt

Last Name

RK Valley

Gender

Male ●

Female ○

Date of Birth

02/02/2002

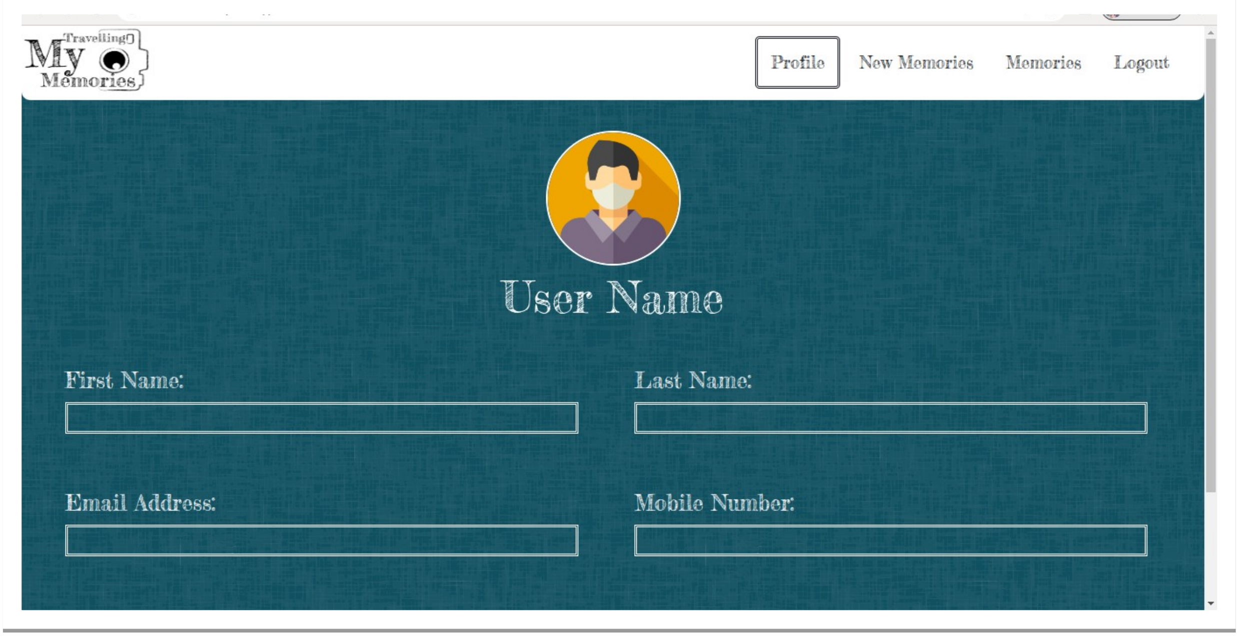
Mobile Number

9988778899

Email

useremail@gmail.com

Profile Page:



User Name

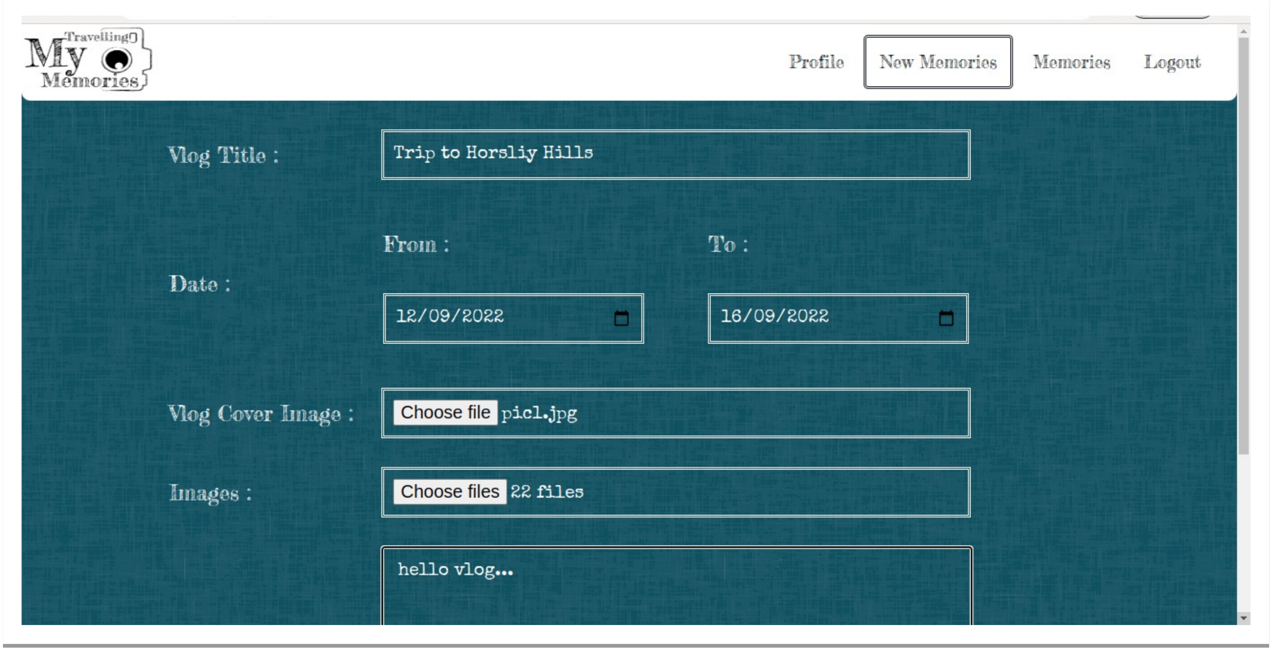
First Name:

Last Name:

Email Address:

Mobile Number:

New Memories:



Vlog Title : Trip to Horsliy Hills

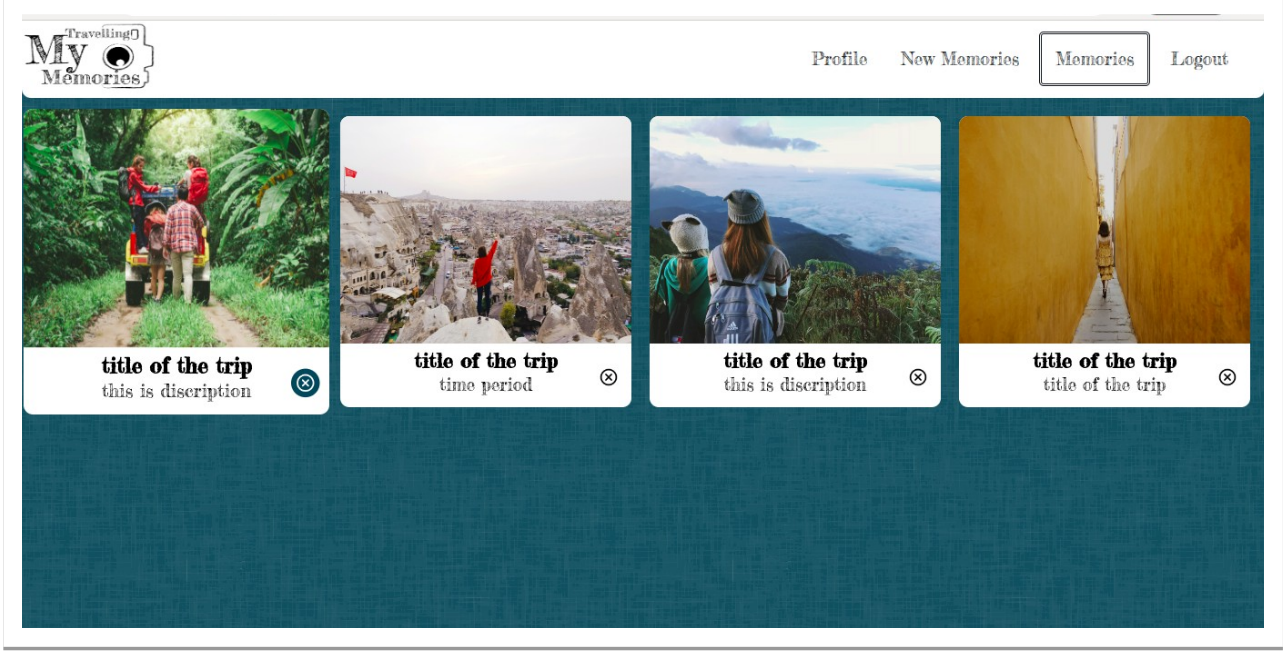
Date : From : 12/09/2022 To : 16/09/2022

Vlog Cover Image : Choose file picl.jpg

Images : Choose files 22 files

hello vlog...

Memories:



title of the trip
this is discription

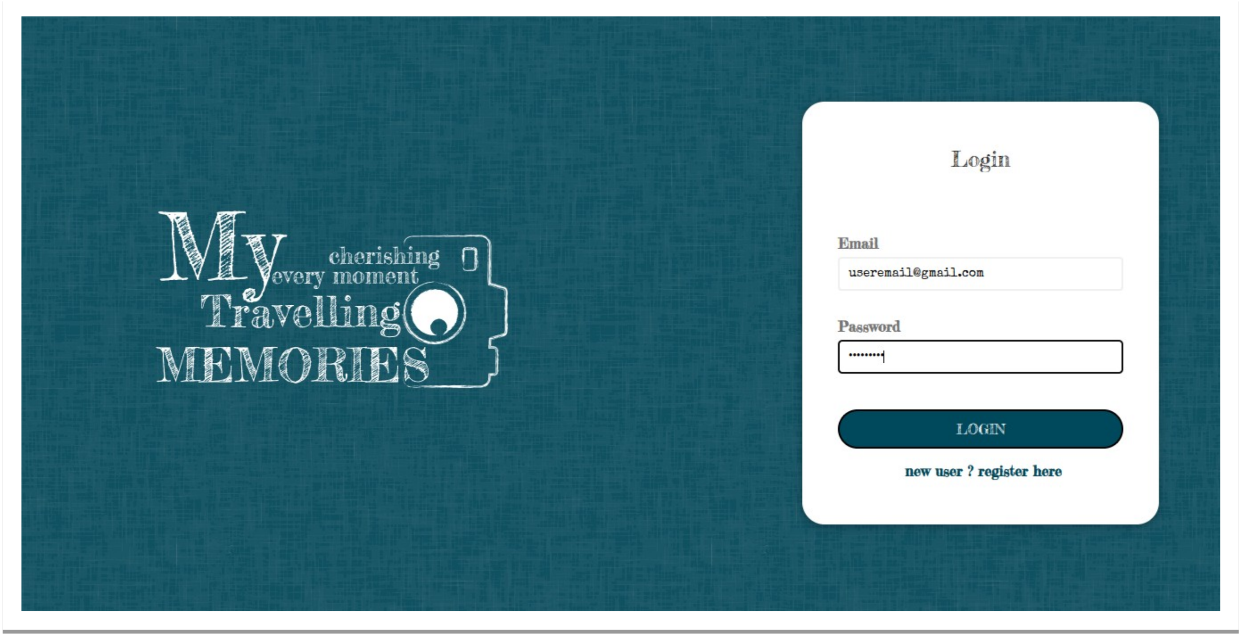
title of the trip
time period

title of the trip
this is discription

title of the trip
title of the trip

Module Testing Snippets

Login Page:



Login

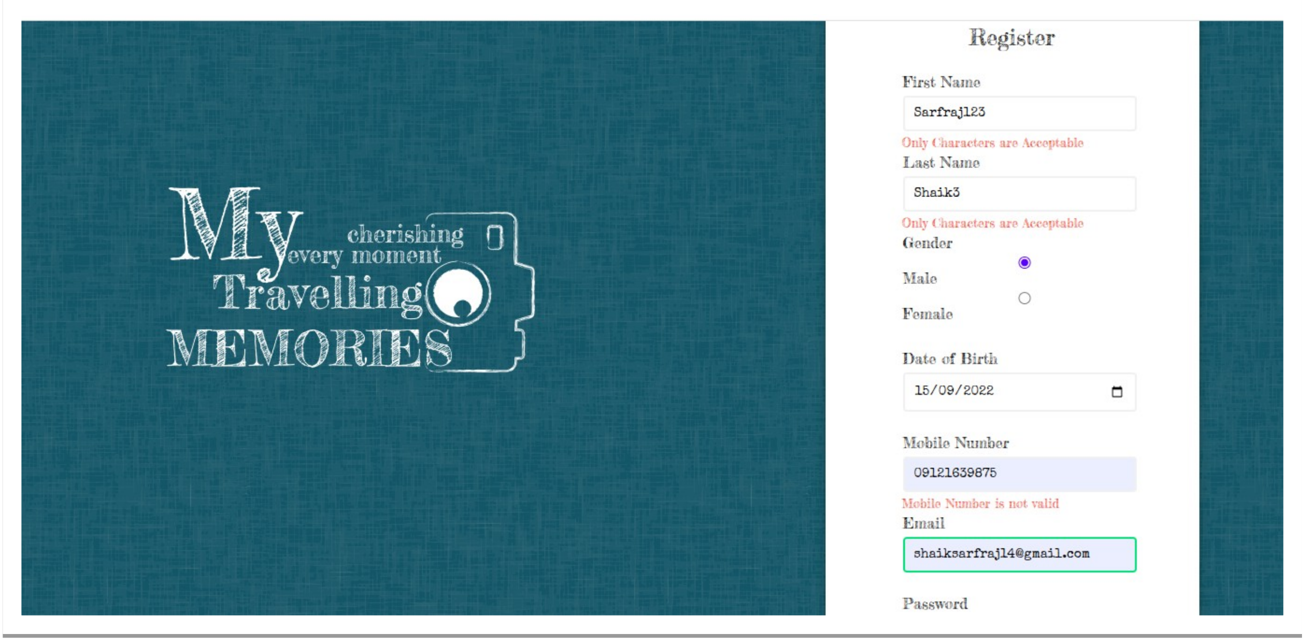
Email

useremail@gmail.com

Password

•••••••••

Register Page:



Register

First Name

Sarfraj123

Only Characters are Acceptable

Last Name

Shaik3

Only Characters are Acceptable

Gender

Male ●

Female ○

Date of Birth

15/09/2022

Mobile Number

09121639875

Mobile Number is not valid

Email

shaiksarfraj14@gmail.com

Password

References

➢https://www.w3schools.com/

➢https://nodejs.org/en/docs/

➢https://expressjs.com/

➢https://www.mongodb.com/