GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

TOURISM WEB APPLICATION

A Project Report of Capstone Project-2

Submitted by

CHANDAN KUMAR

(1613101230)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the Supervision of

Mr. SUBHASH CHANDRA GUPTA

Professor

APRIL / MAY- 2020

GALGOTIAS
UNIVERSITY

SCHOOL OF COMPUTING AND SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report **"TOURISM WEB APPLICATION"** is the bonafide work of **"CHANDAN KUMAR (1613101230)"** who carried out the project work under my supervision.

SIGNATURE OF HEAD	**SIGNATURE OF SUPERVISOR**
Dr. MUNISH SHABARWAL,	Mr. SUBHASH CHANDRA GUPTA,
PhD(Management), PhD(CS)	M.Tech(SE),B.E(CSE)
Professor & Dean,	**Professor,**
School of Computing Science & Engineering	**School of Computing Science & Engineering**

ABSTRACT

As tourism is one of the fastest growing industries today, thus it is very important to have a website or application that can help tourists to know more about the place they want to visit. So I decided to build this project as this can help thousands to lakhs of people who travel. This webapp not only helps to know more about that place but people can also add photos, information related to that place, share their experience of visited places, they can also rate that place, comment something and much more. This project has two main modules, first in which only the verified information will be added by the developers and users here can get all the information about that place like ticket prices, nearest bus stand, nearest railway station, hotels etc. Now the second module and this module is for users to add something to our webapp, they can share their experience and information about the place they visited, they can add photos, ratings ,comments etc. This project is built on AWS cloud9 platform, with technologies like html, css, javascript ,bootstrap for the frontend and nodeJS and expressJS for backend, mongodb as database and heroku for deployment. This is the link to visit the web app https://limitless-everglades-43165.herokuapp.com/ .

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	**iii**
	LIST OF FIGURES	**vi**
1.	**INTRODUCTION**	**1**
	1.1 What is a website?	1
	1.2 What is a web application?	2
	1.3 Why do we need a web application?	2
	1.4 Purpose of choosing this Project	3
	1.5 Existing System	4
	1.6 Proposed System	4
	1.7 Problem Statement	5
	1.8 List of Modules and Usage	5
2.	**SYSTEM REQUIREMENTS**	**7**
	2.1 Hardware Requirements	7
	2.2 Software Requirements	7
3.	**IMPLEMENTATION DIAGRAM**	**8**
4.	**OUTPUTS**	**9**
5.	**SOURCE CODE**	**12**
	5.1 App.js File	12

	5.2 Module Middleware	14
	5.2.1 index.js	14
	5.3 Module models	17
	5.3.1 campgrounds.js	17
	5.3.2 comments.js	18
	5.3.2 review.js	18
	5.3.4 user.js	19
	5.4 Module public	19
	5.4.1 landing.css	19
	5.4.2 main.css	21
	5.5 Module Routes	22
	5.5.1 campgrounds.js	22
	5.5.2 comments.js	25
	5.5.3 index.js	28
	5.5.4 reviews.js	30
	5.6 Module campgrounds	33
	5.6.1 edit.ejs	33
	5.6.2 index.ejs	35
	5.6.3 new.ejs	38
	5.6.4 show.ejs	39
	5.7 Module Partials	43
	5.7.1 header.ejs	44
	5.7.2 footer.ejs	46
6.	**CONCLUSION**	**47**
7.	**REFERENCES**	**48**

LIST OF FIGURES

S.NO.	TITLE	PAGE NO.
1.	Difference b/w website and web application	3
2.	List of modules, files and npm packages used in this project	6
3.	All the pages, buttons, links of the web app and how they works	8
4.	Landing page	9
5.	Home page	9
6.	MoreInfo page	10
7.	Ratings, reviews on MoreInfo page	10
8.	Place added by Us page	11
9.	Users info page	11

1. INTRODUCTION

Tourism can be considered as the most favorite thing when people get free time. Several travel organizations are available on the web where people can book tickets, and on other websites they can see photos of that place, and to get information about that place they visit other websites means there is not a single place where tourists can get all the information at once, and there are no websites present where people can share their experience and information about places they visited, so I decided to make this web application.

This project aimed at developing a web application that will help the people who are going and want to go to a tourist place and want to gather some information about that place like photos, rating, reviews, nearby railway station, airports, hotels, fare of the tickets, luggage keeping facilities etc, and the most important thing is users will also be able to share experience and information about visited places. This web app allows users to add the photos of that place, rate that place, do comments etc. This project can help tourists from all over the world who are going to visit some place. Before going to that place they can gather all the information of that place, this app can be updated in future as the needs of the customers, we can update the app to handle hundreds of users to lakhs of users, in future we can also add features like hotel booking, train and flights tickets booking etc.

1.1 What is a website?

A website is a group of globally accessible, interlinked web pages which have a single domain name. It can be developed and maintained by an individual, business or organization. The website aims to serve a variety of purposes. Example: Blogs. A website is hosted on a single or multiple web server. It is accessible via a network like the Internet or a private local area network via IP address.

1.2 What is a Web Application?

A web application is a software or program which is accessible using any web browser. Its frontend is usually created using languages like HTML, CSS, Javascript, which are supported by major browsers. While the backend could use any programming stack like LAMP, MEAN, etc. Unlike mobile apps, there is no specific SDK for developing web applications.

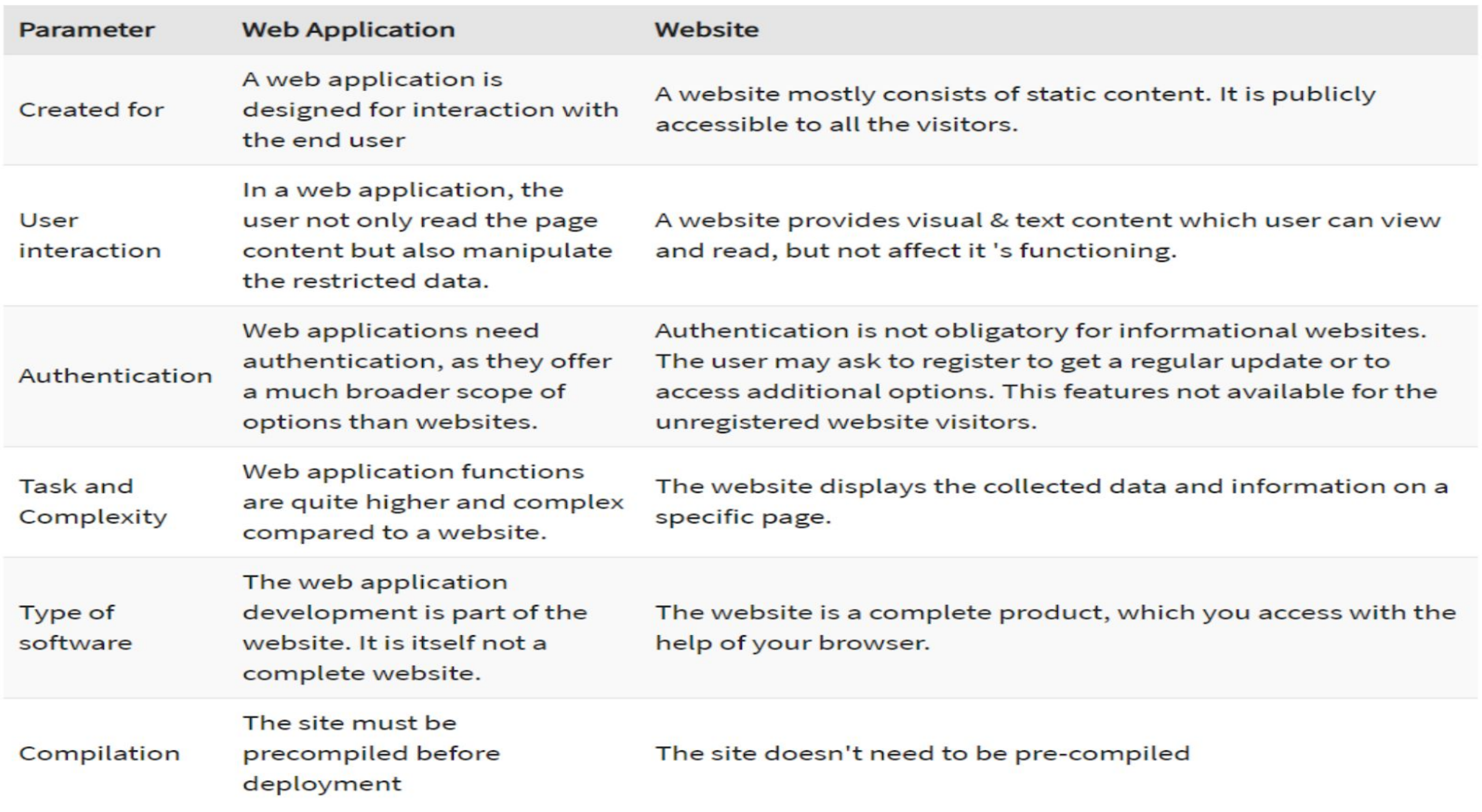
1.3 Why do we need a Web Application?

Web applications are more popular because of the following reasons:

- Compared to desktop applications, web applications are easier to maintain as they use the same code in the entire application. There are no compatibility issues.
- Web applications can be used on any platform: Windows, Linux, Mac… as they all support modern browsers.
- Mobile App store approval not required in web applications.
- Released any time and in any form. No need to remind users to update their applications.
- You can access these web applications 24 hours of the day and 365 days a

year from any PC.

- You can either make use of the computer or your mobile device to access the required data.
- Web applications are a cost-effective option for any organization. Seat Licenses for Desktop software are expensive where SasS are generally, pay as you go.
- Web-Based Apps are Internet-enabled apps that are accessed through the mobile's web browser. Therefore, you aren't required to download or install them.



Parameter	Web Application	Website
Created for	A web application is designed for interaction with the end user	A website mostly consists of static content. It is publicly accessible to all the visitors.
User interaction	In a web application, the user not only read the page content but also manipulate the restricted data.	A website provides visual & text content which user can view and read, but not affect it 's functioning.
Authentication	Web applications need authentication, as they offer a much broader scope of options than websites.	Authentication is not obligatory for informational websites. The user may ask to register to get a regular update or to access additional options. This features not available for the unregistered website visitors.
Task and Complexity	Web application functions are quite higher and complex compared to a website.	The website displays the collected data and information on a specific page.
Type of software	The web application development is part of the website. It is itself not a complete website.	The website is a complete product, which you access with the help of your browser.
Compilation	The site must be precompiled before deployment	The site doesn't need to be pre-compiled

Fig 1: Difference b/w website and web application

1.4 Purpose of choosing this project

I chose this project to help the tourists all around the world, I know what problems tourists face when he goes to a new place, so I have added many features that can help them in getting some information prior going that place like nearest railway station, bus stand, hotels, tickets fare etc, I have also added features so that people can upload photos, give ratings ,reviews, do comments about a visited place. I have used different technologies like html, css, javascript ,bootstrap for the frontend and nodeJS for backend ,mongodb as database and heroku for deployment.

I have divided this app in many modules so that it follows the standards of web development and also followed the REST(representational state transfer) convention for routing this app, and also made basic CRUD(create, read, update, destroy) functionalities, i have also used features like associations and authentication.

1.5 Existing System

Existing web apps and websites do not provide users all the information and functionalities at one place, they either provide booking functionalities, or information related functionality for a particular place, and this is very difficult for the users to visit different websites to book tickets, hotels, and find place related information like nearby places, photos of that place, fare of that place, baggage keeping functionality is present or not, nearest metro, bus or railway stations etc, and all the websites are not mobile phones compatible and they also do not provide chat bots for asking queries.

1.6 Proposed System

Our web app have all the basic features that a website contains, like login, signup, profile of the user, authentication features, search bar, landing page, besides that we also added links to other popular websites for convenience of the users like facebook, youtube, twitter, instagram, chatbot for asking queries and assistance. And most important functionality which is not provided by other websites like, our users are able to share their experience about visited places and they can add photos of that place, edit them, do comments, rate and review a place etc.

1.7 Problem Statement

Going to a new place as a tourist is not so easy, we do lots of effort before going to a new place like we gather some useful information about that place like, location, ticket fare, how to reach that place, where to stay, how much time we need to go and come back, how much money we require, which hotels are good in that area, nearby places to visit etc.

I also faced many problems when I was doing this project like which tools I should use, which technologies I should use, which programming language for backend, which framework for frontend, from where to learn to use these technologies. What features I should add in this webapp.

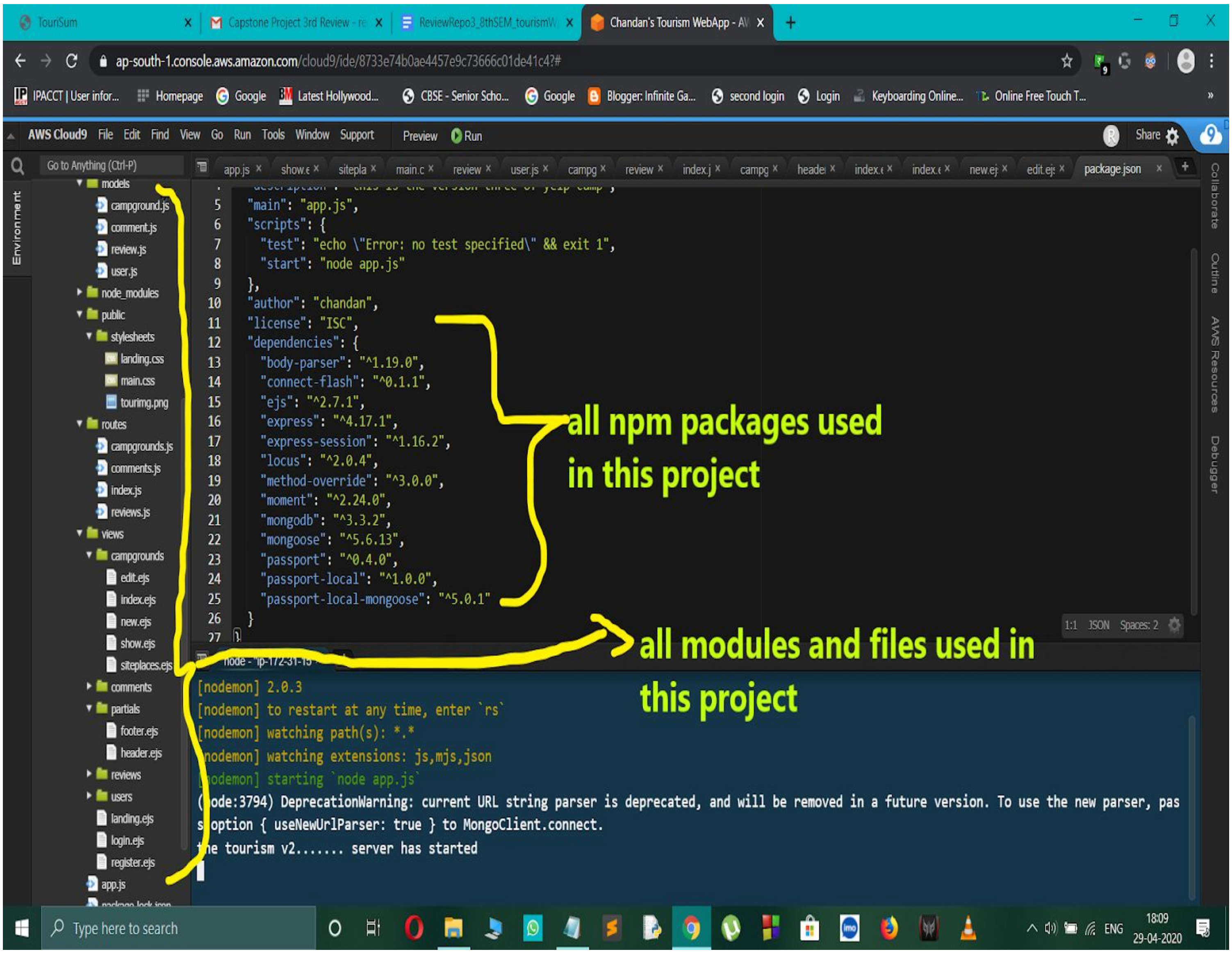
1.8 List of modules and usage

Views: This module contains all the express files for different routes.
Node Modules: This contains all the node packages that we need to develop our app.
Public: This contains all the stylesheets to style our app.

Routes: This contains all the different routes we will need in our app, Ex-user, add places, comments, login, signup, update, show info etc.

Middleware: This will contain all the code that we need to run before some event occurs on our app.

Our end product is a fully functional web application that can be accessed by the user from anywhere on any device and on any network, this app can also be used on mobiles.



```
  "main": "app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node app.js"
  },
  "author": "chandan",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.19.0",
    "connect-flash": "^0.1.1",
    "ejs": "^2.7.1",
    "express": "^4.17.1",
    "express-session": "^1.16.2",
    "locus": "^2.0.4",
    "method-override": "^3.0.0",
    "moment": "^2.24.0",
    "mongodb": "^3.3.2",
    "mongoose": "^5.6.13",
    "passport": "^0.4.0",
    "passport-local": "^1.0.0",
    "passport-local-mongoose": "^5.0.1"
  }
}
```

```
[nodemon] 2.0.3
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
(node:3794) DeprecationWarning: current URL string parser is deprecated, and will be removed in a future version. To use the new parser, pass option { useNewUrlParser: true } to MongoClient.connect.
the tourism v2....... server has started
```

Fig 2: List of modules, files and npm packages used in this project

2. SYSTEM REQUIREMENTS

This chapter tells which hardware, softwares and languages we needed to make this web application.

2.1 Hardware Requirements

Processor(Core 2 Duo or above).

RAM(2gb or above).

HDD(100gb or above).

2.2 Software Requirements

Environment used: AWS cloud9.

Languages and Technologies Used to build this application:

Frontend: Html 5, CSS 3, Javascript, Bootstrap, Jquery.

Backend: Node JS, Express, Mongoose.

Database: MongoDB.

Version Control System: Git.

Heroku: For deployment.

3. IMPLEMENTATION DIAGRAM

This chapter helps you to know about how this web application works, on clicking which button you go where, and it also has all the pages that our app contains.

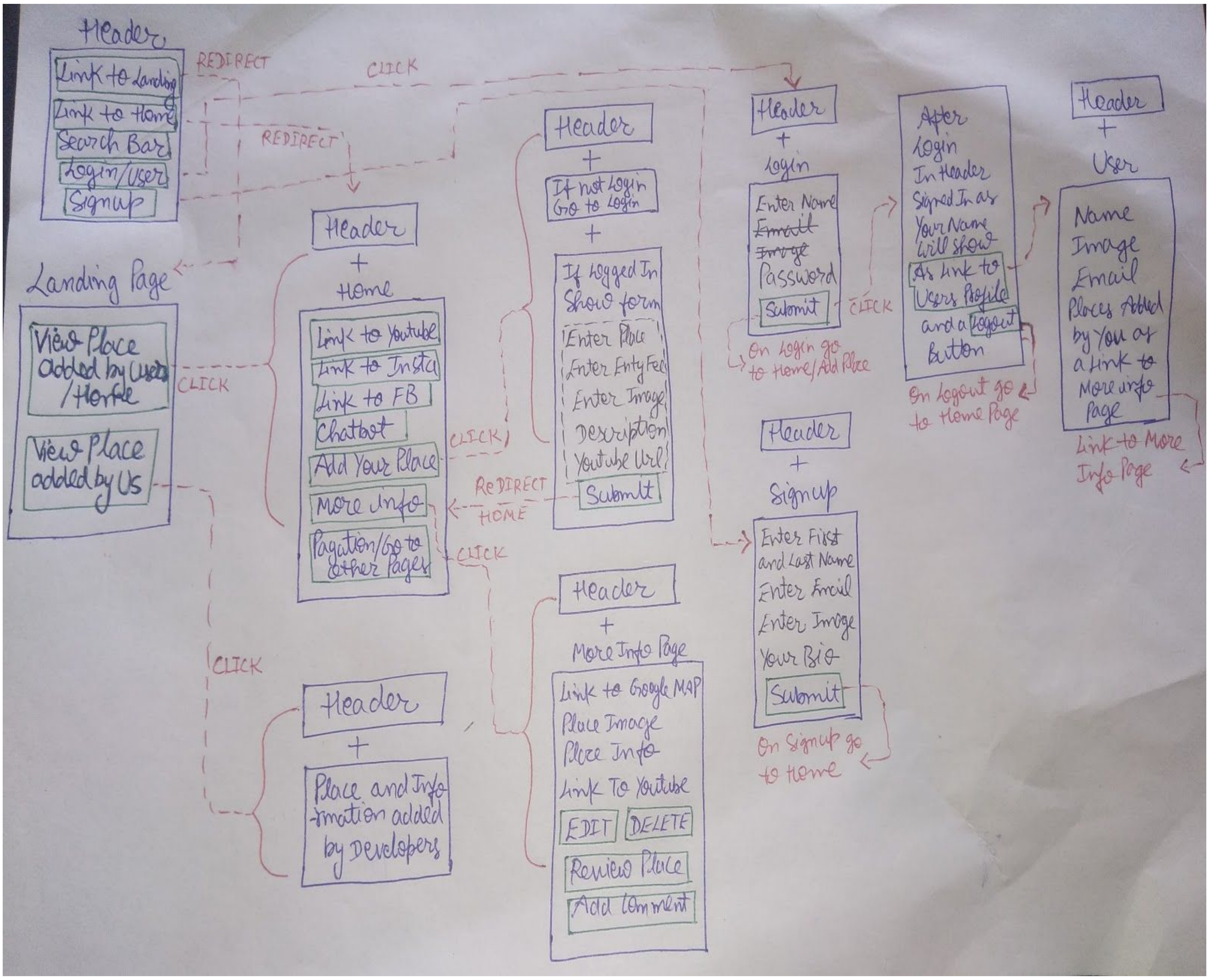


Fig 3: All the pages, buttons, links of the web app and how they works

4. OUTPUTS

This chapter contains the images of different pages that our app contains.

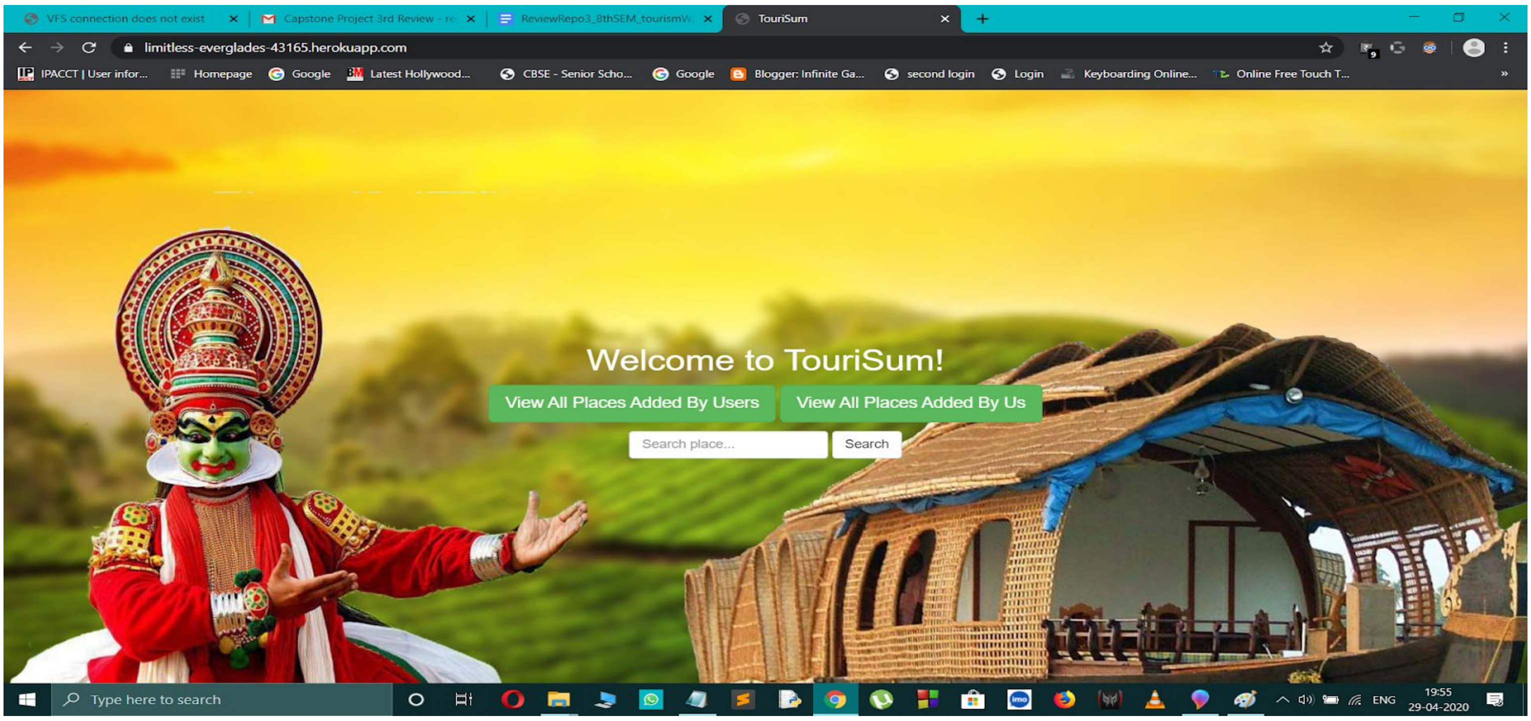
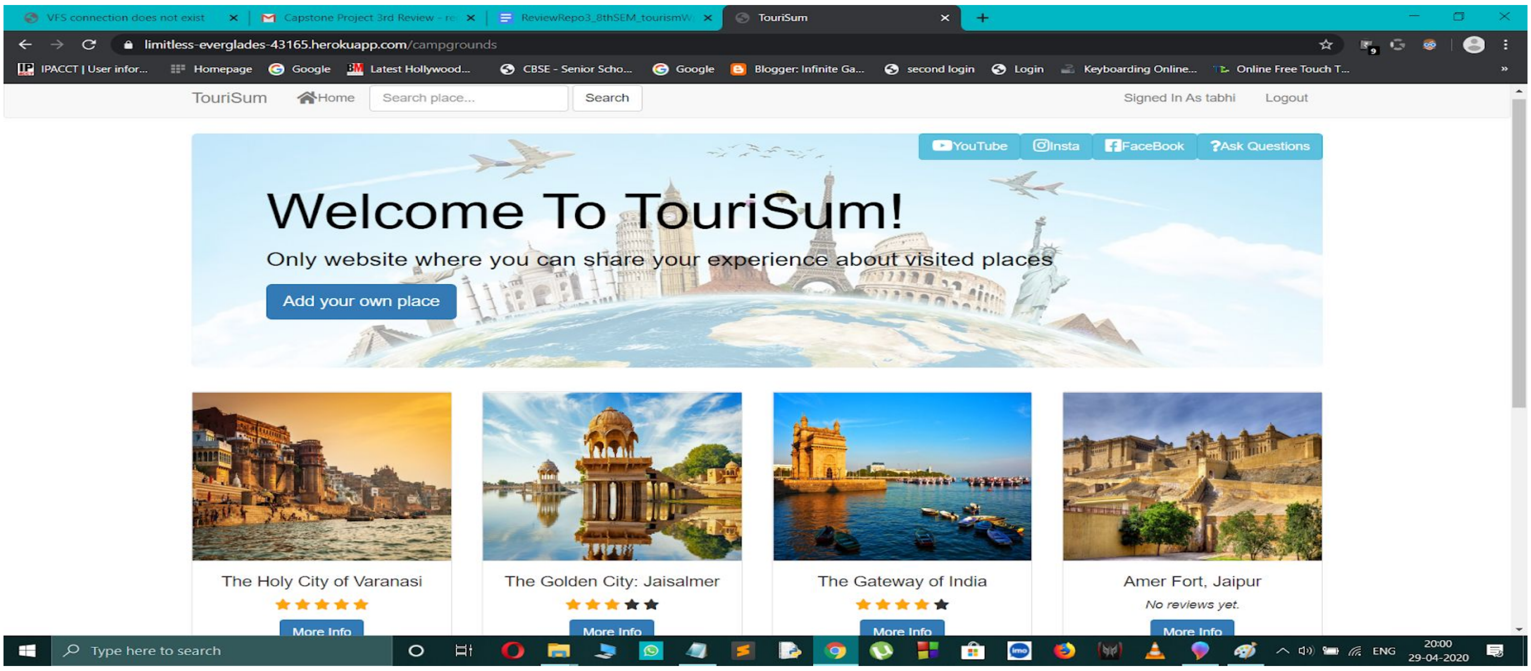

limitless-everglades-43165.herokuapp.com
Welcome to TouriSum!
View All Places Added By Users
View All Places Added By Us
Search place...
Search

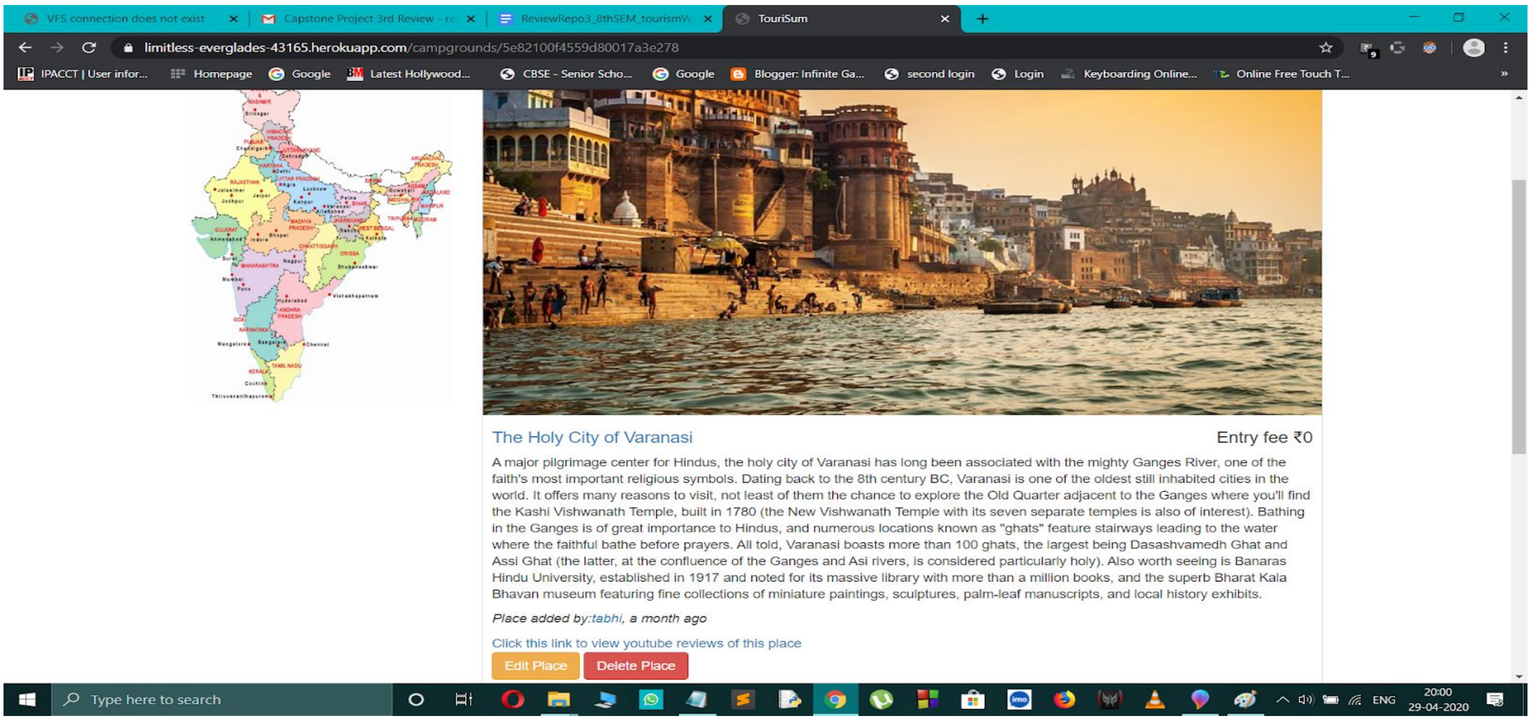
Fig 4: Landing page



Welcome To TouriSum!

Only website where you can share your experience about visited places

Fig 5: Home page


TouriSum
limitless-everglades-43165.herokuapp.com/campgrounds/5e82100f4559d80017a3e278
Type here to search

The Holy City of Varanasi

Entry fee ₹0

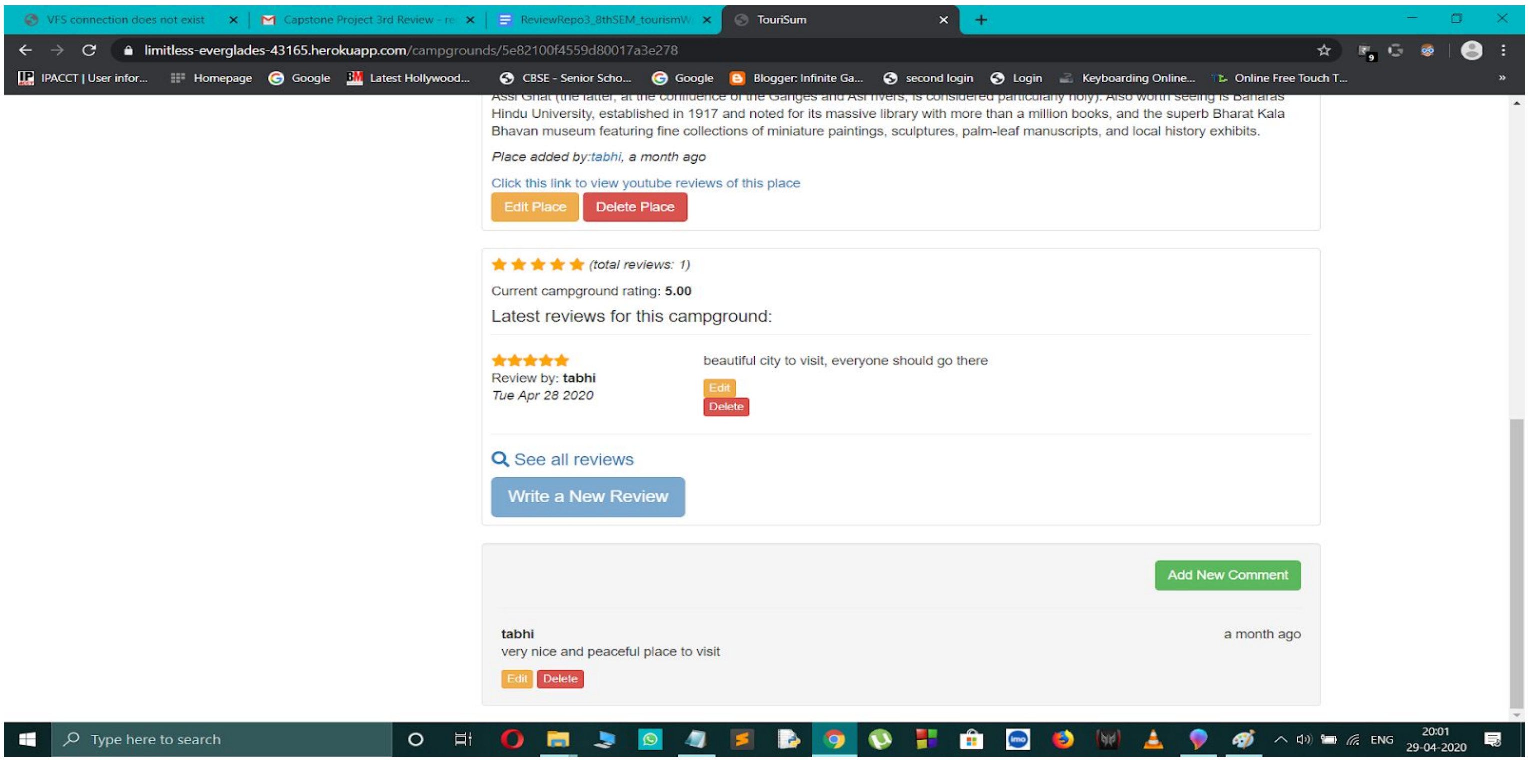
A major pilgrimage center for Hindus, the holy city of Varanasi has long been associated with the mighty Ganges River, one of the faith's most important religious symbols. Dating back to the 8th century BC, Varanasi is one of the oldest still inhabited cities in the world. It offers many reasons to visit, not least of them the chance to explore the Old Quarter adjacent to the Ganges where you'll find the Kashi Vishwanath Temple, built in 1780 (the New Vishwanath Temple with its seven separate temples is also of interest). Bathing in the Ganges is of great importance to Hindus, and numerous locations known as "ghats" feature stairways leading to the water where the faithful bathe before prayers. All told, Varanasi boasts more than 100 ghats, the largest being Dasashvamedh Ghat and Assi Ghat (the latter, at the confluence of the Ganges and Asi rivers, is considered particularly holy). Also worth seeing is Banaras Hindu University, established in 1917 and noted for its massive library with more than a million books, and the superb Bharat Kala Bhavan museum featuring fine collections of miniature paintings, sculptures, palm-leaf manuscripts, and local history exhibits.

Place added by:tabhi, a month ago

Click this link to view youtube reviews of this place

Edit Place Delete Place

Fig 6: MoreInfo page


TouriSum
limitless-everglades-43165.herokuapp.com/campgrounds/5e82100f4559d80017a3e278
Type here to search

Hindu University, established in 1917 and noted for its massive library with more than a million books, and the superb Bharat Kala Bhavan museum featuring fine collections of miniature paintings, sculptures, palm-leaf manuscripts, and local history exhibits.

Place added by:tabhi, a month ago

Click this link to view youtube reviews of this place

Edit Place Delete Place

★★★★★ *(total reviews: 1)*

Current campground rating: **5.00**

Latest reviews for this campground:

★★★★★
Review by: **tabhi**
Tue Apr 28 2020

beautiful city to visit, everyone should go there

Edit
Delete

See all reviews

Write a New Review

Add New Comment

tabhi a month ago

very nice and peaceful place to visit

Edit Delete

Fig 7: Rating, Reviews and comments on MoreInfo page

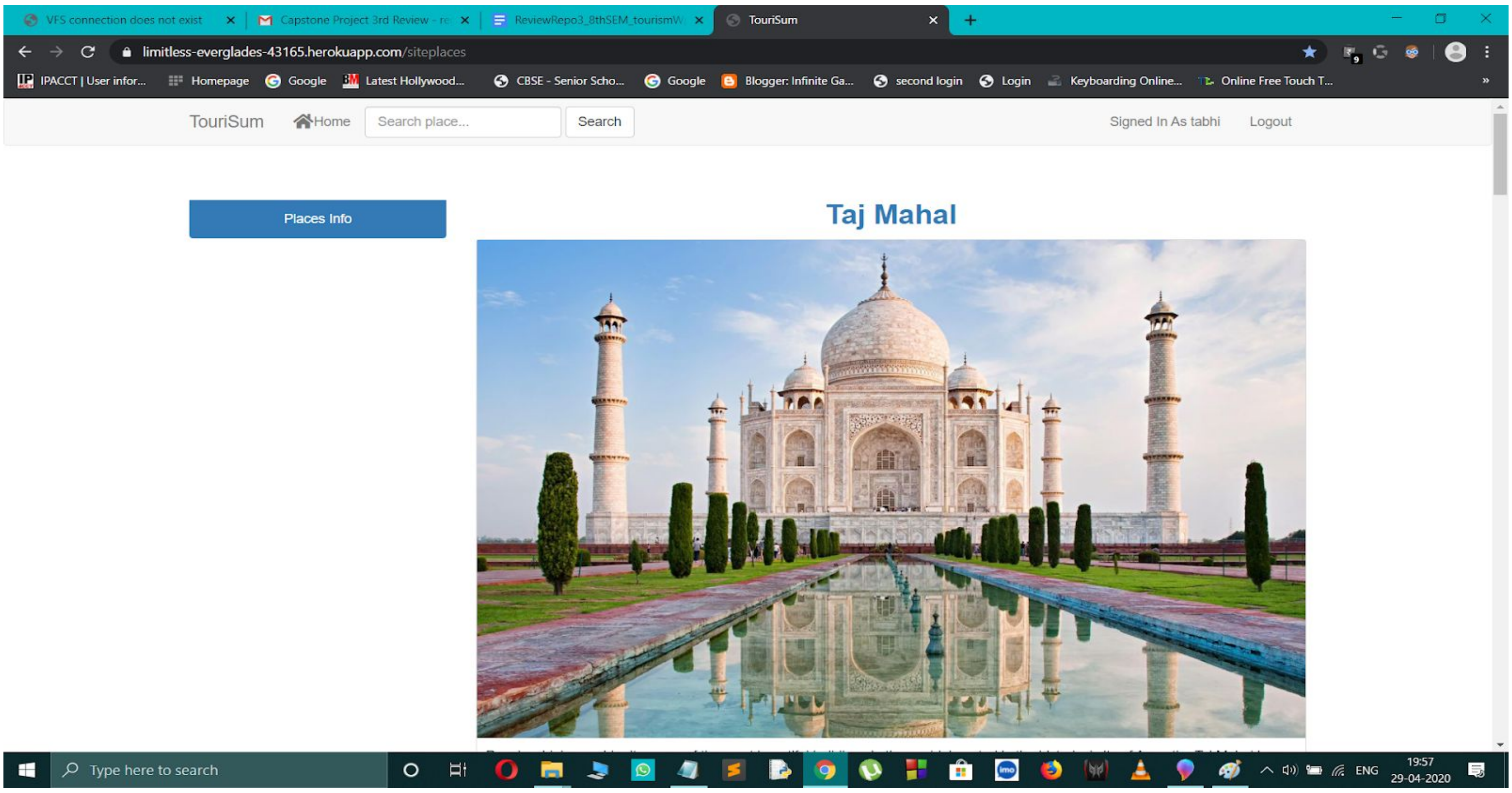
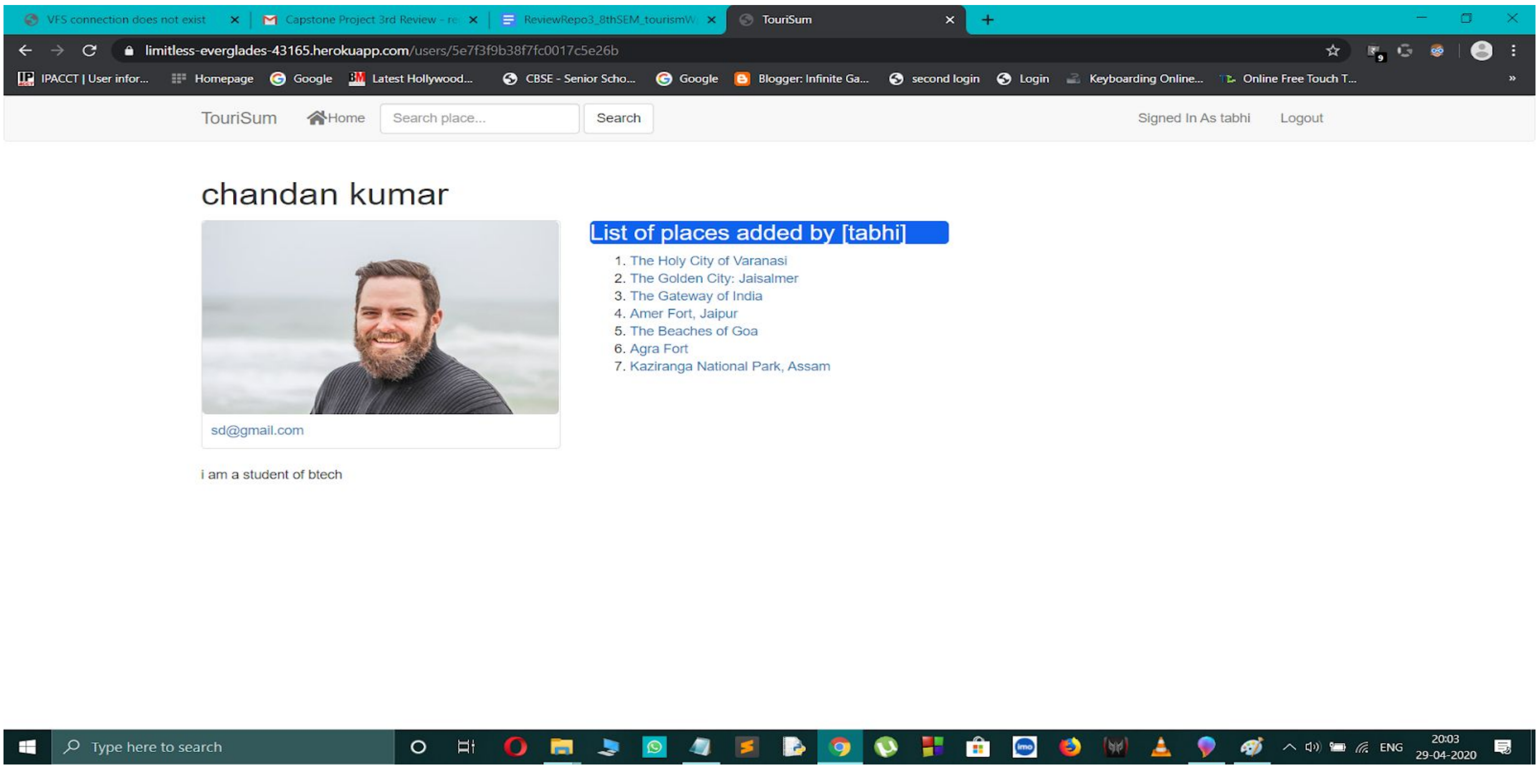


Fig 8: Places added by Us page


limitless-everglades-43165.herokuapp.com/users/5e7f3f9b38f7fc0017c5e26b
TouriSum
Home
Search place...
Search
Signed In As tabhi
Logout

chandan kumar

sd@gmail.com

i am a student of btech

List of places added by [tabhi]

1. The Holy City of Varanasi
2. The Golden City: Jaisalmer
3. The Gateway of India
4. Amer Fort, Jaipur
5. The Beaches of Goa
6. Agra Fort
7. Kaziranga National Park, Assam

Fig 9: Users info page

5. SOURCE CODE

This chapter contains the source code for the whole web app, our web app has many modules and each module contains code for different purposes. App.js is the file from where our app will start because this file binds all the modules together. Different modules that present in our web app are: Middleware, Public, Routes, Views and this module further contains different sub modules namely Campgrounds, Partials, Comments, Reviews, Users.

5.1 App.js File

This is the file that is needed to bind all the modules together to make them work.

```
var express          = require("express"),
app             = express(),
bodyParser       = require("body-parser"),
mongoose         = require("mongoose"),
flash            = require("connect-flash"),
passport         = require("passport"),
localStrategy    =require("passport-local"),
methodOverride   = require("method-override"),
 Campground       = require("./models/campground"),
 Comment          = require("./models/comment"),
 User             = require("./models/user"),
 seedDB           = require("./seeds")
 //requiring routes
 var commentRoutes    = require("./routes/comments") ,
 reviewRoutes     = require("./routes/reviews"),
 campgroundRoutes = require("./routes/campgrounds"),
indexRoutes      = require("./routes/index")
Var Url= process.env.DATABASEURL ||"mongodb://localhost/tourism_v2"
app.use(bodyParser.urlencoded({extended: true}))
app.set("view engine","ejs")
app.use(express.static(__dirname+"/public"))
app.use(methodOverride("_method"))
app.use(flash())
app.locals.moment = require('moment');
//PASSPORT CONFIGURATION
app.use(require("express-session")({ secret: "once again rusty wins the cutest
dog award",  resave: false, saveUninitialized:false }))
app.use(passport.initialize())
app.use(passport.session())
passport.use(new localStrategy(User.authenticate()))
passport.serializeUser(User.serializeUser())
passport.deserializeUser(User.deserializeUser())
app.use(function(req , res, next){ res.locals.currentUser = req.user
res.locals.error = req.flash("error")
res.locals.success = req.flash("success")
next() })
//this is the usage of express-router
app.use("/",indexRoutes)
app.use("/campgrounds",campgroundRoutes)
app.use("/campgrounds/:id/comments",commentRoutes)
app.use("/campgrounds/:id/reviews", reviewRoutes);
app.get("/siteplaces",function(req,res){
res.render("campgrounds/siteplaces") })
```

5.2 Module Middleware

This module contains code that is used so that we can run some code before executing the other code.

5.2.1 Index.js

This file contains code for all the middleware used in our app.

```
var Campground = require("../models/campground")
var Comment     = require("../models/comment")
var Review = require("../models/review");
//all the middleware codes
var middlewareObj = {}
middlewareObj.checkCampgroundOwnership = function(req ,res ,next){
if(req.isAuthenticated()){
Campground.findById(req.params.id, function(err,foundCampground){
if(err || !foundCampground){
req.flash("error","Place not found")
res.redirect("back")
} else{
if(foundCampground.author.id.equals(req.user._id)){
next()
}  else{
req.flash("error", "you dont have permission to do that")
res.redirect("back") }}})
} else{
req.flash("error","You need to be loggedIn to do that")
res.redirect("back")      }}
```

```
middlewareObj.checkCommentOwnership = function(req ,res ,next){
if(req.isAuthenticated()){
Comment.findById(req.params.comment_id, function(err,foundComment){
if(err || !foundComment){
req.flash("error","Comment not found")
res.redirect("back")
} else{
if(foundComment.author.id.equals(req.user._id)){                next()
}  else{
req.flash("error","You dont have permission to do that" res.redirect("back")
}}})
} else{
req.flash("error","You need to be loggedIn to do that")
res.redirect("back")  } }
middlewareObj.checkReviewOwnership = function(req, res, next) {
if(req.isAuthenticated()){
Review.findById(req.params.review_id, function(err, foundReview){
if(err || !foundReview){
res.redirect("back");
}  else {
if(foundReview.author.id.equals(req.user._id)) {
next();
} else {
req.flash("error", "You don't have permission to do that");
res.redirect("back") }} });
} else {
req.flash("error", "You need to be logged in to do that");
res.redirect("back"); }};
middlewareObj.checkReviewExistence = function (req, res, next) {
if (req.isAuthenticated(
Campground.findById(req.params.id).populate("reviews").exec(function (err,
foundCampground) {
if (err || !foundCampground) {
req.flash("error", "Campground not found.");
res.redirect("back");
} else{ var foundUserReview = foundCampground.reviews.some(function
(review) { return review.author.id.equals(req.user._id) });
if (foundUserReview) {
req.flash("error", "You already wrote a review.");
return res.redirect("/campgrounds/" + foundCampground._id);
}  next()  } });
} else {
req.flash("error", "You need to login first.");
res.redirect("back")  } };
middlewareObj.isLoggedIn = function(req,res, next){
if(req.isAuthenticated()){
return next() }
req.flash("error","You need to be loggedIn to do that")
res.redirect("/login") }
module.exports= middlewareObj;
```

5.3 Module Models

This module contains the code for different types of schema, that is used to make different types of models like campgrounds, comments, reviews, users etc. And these models further help to store and retrieve data from the database.

5.3.1 Campgrounds.js

This file contains the code for different places and their model like what type of data they contain, name, image, description, comments, reviews etc.

```
var mongoose = require("mongoose");
var campgroundSchema = new mongoose.Schema({
name:String, price:String, image:String, description:String,
createdAt: { type: Date, default: Date.now },
author: {
id: {
type: mongoose.Schema.Types.ObjectId,
ref:  "User” },
username:String },
Youtubeurl :String,
Comments:[{ type:mongoose.Schema.Types.ObjectId, ref:"Comment" }],
reviews: [ { type: mongoose.Schema.Types.ObjectId,  ref: "Review" } ],
rating: { type: Number , default: 0  }})
module.exports = mongoose.model("Campground" , campgroundSchema)
```

5.3.2 Comments.js

This file contains the code for different comments and their model like what type of data they contain, text, created at, author.

```
var mongoose = require("mongoose")
var commentSchema = mongoose.Schema({
text: String,
createdAt: { type: Date, default: Date.now },
author: {
id: {  type:mongoose.Schema.Types.ObjectId, ref:"User"    } ,
username:String  }})
module.exports = mongoose.model("Comment",commentSchema)
```

5.3.3 Review.js

This file contains the code for different reviews and their model like what type of data they contain, rating, text, author etc.

```
var mongoose = require("mongoose");
var reviewSchema = new mongoose.Schema({
rating: {
type: Number,        required: "Please provide a rating (1-5 stars).",
min: 1,  max: 5,
validate: {
validator: Number.isInteger,
message: "{VALUE} is not an integer value."
}},
text: {  type: String },
author: {
```

```
id: { type: mongoose.Schema.Types.ObjectId,  ref: "User" },
username: String }
campground: {
type: mongoose.Schema.Types.ObjectId,
ref: "Campground"
} },{ timestamps: true });
module.exports = mongoose.model("Review", reviewSchema);
```

5.3.4 User.js

This file contains the code for different users and their model like what type of data they contain, name, password, image, description, comments, reviews etc.

```
var mongoose = require("mongoose")
var passportLocalMongoose = require("passport-local-mongoose")
var UserSchema = new mongoose.Schema({
username: String, password: String, avatar:   String, firstName:String,
lastName:String, email:String, description:String })
UserSchema.plugin(passportLocalMongoose)
module.exports = mongoose.model("User",UserSchema)
```

5.4 Module Public

This Module contains the code for different places and how to style them, basically it contains CSS stylesheets etc.

5.4.1 Landing.css

This file contains the styling code for the landing page of the web app.

```
body { background-color: #000; }
#landing-header {   z-index: 1; position: relative; text-align: center;
padding-top: 40vh; }
#landing-header h1 {color: #fff; }
.slideshow {   position: fixed; width: 100%; height: 100%; top: 0; left: 0;
z-index: 0; list-style: none; margin: 0; padding: 0;}
.slideshow li {  width: 100%; height: 100%; position: absolute; top: 0;  left: 0;
background-size: cover;  background-position: 50% 50%;
background-repeat: no-repeat;  opacity: 0;  z-index: 0;
animation: imageAnimation 50s linear infinite; }
.slideshow                                        li:nth-child(1){background-image:
url(https://timebusinessnews.com/wp-content/uploads/1733_1-2.jpg) }
.slideshow li:nth-child(2) {
background-image:
url(https://blog.hubspot.com/hubfs/Design%20101%20Asymmetrical%20and%
20Symmetrical%20Balance.png);
animation-delay: 10s; }
.slideshow li:nth-child(3) {
background-image: url(https://wallpapercave.com/wp/n2paec5.jpg);
animation-delay: 20s; }
```

```
.slideshow li:nth-child(4) {
background-image:
url(https://www.jakpost.travel/wimages/large/22-224304_full-hd-wallpaper-boa
t-beach-azure-ocean-goa.jpg);
animation-delay: 30s; }
.slideshow li: nth-child(5) {background-image:
url(https://upload.wikimedia.org/wikipedia/commons/b/b5/Vishnath_temple.pn
g
animation-delay: 40s; }
@keyframes imageAnimation {
0% {  opacity: 0;  animation-timing-function: ease-in;}
10% { opacity: 1; animation-timing-function: ease-out;}
20% { opacity: 1}
30% {opacity: 0}}
.no-cssanimations .slideshow li {
opacity: 1; }
```

5.4.2 Main.css

This file contains the code for stylesheets which is used to style pages other than landing.

```
.thumbnail img { width:100%;  height: 220px; background-size: cover;
background-position: center; }
.thumbnail {  padding:0; }
.thumbnail .caption-full { padding:9px;}
.delete-form{ display: inline }
.thumb > img{ width:100%; height:550px; }
```

```
.checked { color: orange;}
.jumbotron{
background: url("tourimg.png") no-repeat center center ;
-webkit-background-size: 100% 100%;
-moz-background-size: 100% 100%;
-o-background-size: 100% 100%;
background-size: 100% 100%; }
#map { height: 400px; width: 100%; },.cmap:hover img { opacity:.7;}
```

5.5 Module Routes

This Module contains the code for different routes, means for different other pages and how we reach them.

5.5.1 Campgrounds.js

This file contains the route code for different places.

```
var express = require("express")
var router  = express.Router()
var Campground =require("../models/campground")
var middleware= require("../middleware")
var Review = require("../models/review");
var Comment     = require("../models/comment")
router.get("/", function(req, res){    var perPage = 8;
var pageQuery = parseInt(req.query.page);
var pageNumber = pageQuery ? pageQuery : 1;
var noMatch = null;
if(req.query.search)     {                                    const     regex     =     new
RegExp(escapeRegex(req.query.search), 'gi');
Campground.find({name:     regex}).skip((perPage     *     pageNumber)     -
perPage).limit(perPage).exec(function (err, allCampgrounds) {
Campground.count({name: regex}).exec(function (err, count) {
if (err) { console.log(err); res.redirect("back");
} else {
if(allCampgrounds.length < 1) {
```

```
noMatch = "No campgrounds match that query, please try again."; }
res.render("campgrounds/index", {
campgrounds: allCampgrounds,
current: pageNumber,
pages: Math.ceil(count / perPage),
noMatch: noMatch,
search: req.query.search
});} }); });
} else{ Campground.find({}).skip((perPage * pageNumber) - perPage).limit(perPage).exec(function (err, allCampgrounds) {
Campground.count().exec(function (err, count) {
if (err) { console.log(err);
} else { res.render("campgrounds/index", {
campgrounds: allCampgrounds,current: pageNumber,
pages: Math.ceil(count / perPage,noMatch: noMatch,
search: false }); }});})}});
//CREATE - add new campground to DB
router.post("/",middleware.isLoggedIn,function(req,res){
var name= req.body.name
var price = req.body.price
var image= req.body.image
var desc = req.body.description
var author ={
id: req.user._id,
username: req.user.username}
var youtubeurl = req.body.youtubeurl
var newCampground = {name: name, price:price, image: image,
description:desc, author:author, youtubeurl:youtubeurl }
//create a new campground and save to the DB
Campground.create(newCampground,function(err,newlyCreated){
if(err){console.log(err) } else{  console.log(newlyCreated)
res.redirect("/campgrounds") }})})
//NEW- show form to create new campground
router.get("/new",middleware.isLoggedIn,function(req,res){
res.render("campgrounds/new")})
//SHOW - shows more info about one campground
router.get("/:id",function(req,res){
Campground.findById(req.params.id).populate("comments").populate({
Path:"reviews", options: {sort: {createdAt: -1}}
}).exec(function(err,foundCampground){
if(err || !foundCampground){ req.flash("error","Campground not found")
res.redirect("back") } else{
res.render("campgrounds/show",{ campground: foundCampground}) }})})
router.get("/:id/edit", middleware.checkCampgroundOwnership,function(req,
res{ Campground.findById(req.params.id, function(err,foundCampground){
res.render("campgrounds/edit",{campground:foundCampground})})})
router.put("/:id",middleware.checkCampgroundOwnership, function(req, res)
Campground.findByIdAndUpdate(req.params.id, req.body.campground,
function(err, updatedCampground){ if(err){ res.redirect("/campgrounds"
} else{ res.redirect("/campgrounds/" + req.params.id) }})})
router.delete("/:id", middleware.checkCampgroundOwnership, function (req,
res) { Campground.findById(req.params.id, function (err, campground) {
if (err) { res.redirect("/campgrounds");
} else { Comment.remove({"_id": {$in: campground.comments}},
function (err) {
if (err) { console.log(err);
return res.redirect("/campgrounds"); }
Review.remove({"_id": {$in: campground.reviews}}, function (err) {
if (err) { console.log(err); return res.redirect("/campgrounds");}
campground.remove();
req.flash("success", "Campground deleted successfully!");
res.redirect("/campgrounds");
});});}});});
function escapeRegex(text) {
return text.replace(/[-[\]{}()*+?.,\\^$|#\s]/g,"\\$&")}
module.exports = router
```

5.5.2 Comments.js

This file contains the route code for different comments.

```
var express     = require("express")
var router      = express.Router({mergeParams : true})
var Campground  = require("../models/campground")
var Comment     = require("../models/comment")
var middleware= require("../middleware")
//comments new
router.get("/new",middleware.isLoggedIn ,function(req, res) {          //here
the whole route is /campgrounds/:id/comments/new , but we have shorten them
using code on line 59 in app.js file
//find campground by id
```

```
Campground.findById(req.params.id , function(err,campground){
if(err){
console.log(err)
} else{
res.render("comments/new",{campground:campground})
} })})
//comments create
router.post("/",            middleware.isLoggedIn            ,            function(req,res){
Campground.findById(req.params.id, function(err, campground) {
if(err){
console.log(err)
res.redirect("/campgrounds")
} else{
Comment.create(req.body.comment, function(err,comment){
if(err){
req.flash("error","Something went wrong")
console.log(err)
} else{
//add username and id to comment
comment.author.id = req.user._id
comment.author.username = req.user.username
//save comments
comment.save()
campground.comments.push(comment)
campground.save()
req.flash("success","Successfully added comment")
res.redirect('/campgrounds/' + campground._id)
}})}})})
//Comments Edit Route
router.get("/:comment_id/edit",middleware.checkCommentOwnership,
function(req,res){                Campground.findById(req.params.id,    function(err,
foundCampground){
if(err                               ||                               !foundCampground){
req.flash("error","No Campground found")
return res.redirect("back") }})
Comment.findById(req.params.comment_id, function(err, foundComment) {
if(err){ res.redirect("back") } else{
res.render("comments/edit",{campground_id:req.params.id,
comment:foundComment})}})})
//COMMENT UPDATE
router.put("/:comment_id",middleware.checkCommentOwnership,
function(req,res){
Comment.findByIdAndUpdate(req.params.comment_id,       req.body.comment,
function(err, updatedComment){       if(err){
res.redirect("back")
} else{ res.redirect("/campgrounds/" + req.params.id) }})})
//COMMENT DESTROY ROUTE
router.delete("/:comment_id",middleware.checkCommentOwnership,
function(req,res){
Comment.findByIdAndRemove(req.params.comment_id, function(err){
if (err) {
res.redirect("back")
} else{
req.flash("success","Comment deleted")
```

```
res.redirect("/campgrounds/" + req.params.id) }})})
```

5.5.3 Index.js

This file contains the route code for different places.

```
var express     = require("express")
var router      = express.Router()
var passport    = require("passport")
var User        = require("../models/user")
var Campground = require("../models/campground")
router.get("/",function(req,res){
res.render("landing")})
//show register form
router.get("/register",function(req, res) {
res.render("register") })
//handle sign up logic
router.post("/register",function(req,res){
var newUser= new User({
username: req.body.username,firstName: req.body.firstName,
lastName: req.body.lastName,email: req.body.email,
avatar: req.body.avatar,description: req.body.description })
User.register( newUser , req.body.password , function(err,user){
if(err){ console.log(err);
res.render('register',{error:err.message});                    }
passport.authenticate("local")(req,res, function(){
req.flash("success","Welcome to TouriSum " + user.username)      //it will print
message with the name of the user
res.redirect("/campgrounds") })})})
//show login form
router.get("/login",function(req, res) { res.render("login")  })
app.post("/login", middleware ,callback)
router.post("/login", passport.authenticate("local",{
successRedirect:"/campgrounds",
failureRedirect:"/login"   }) , function(req,res)} })
//logout route
router.get("/logout", function(req, res) {
req.logout()
req.flash("success", "Logged you out!")
res.redirect("/campgrounds") })
// user profile
router.get("/users/:id",function(req, res) {
User.findById(req.params.id,function(err,foundUser){
if(err){ req.flash("error","Something went wrong")
return res.redirect("/"); }
Campground.find().where("author.id").equals(foundUser._id).exec(function(err,
campgrounds){
if(err){ req.flash("error","Something went wrong")
return res.redirect("/");
res.render("users/show",{user:foundUser,campgrounds: campgrounds})})})})})
module.exports = router
```

5.5.4 Reviews.js

This file contains the route code for different Reviews.

```
var express = require("express");
var router = express.Router({mergeParams: true});
var Campground = require("../models/campground"
var Review = require("../models/review");
var middleware = require("../middleware");
// Reviews Index
router.get("/", function (req, res) {
Campground.findById(req.params.id).populate({
path: "reviews",
options: {sort: {createdAt: -1}}     }).exec(function (err, campground) {
if (err || !campground) {
req.flash("error", err.message);
return res.redirect("back"); }
res.render("reviews/index", {campground: campground});});});
// Reviews New
router.get("/new", middleware.isLoggedIn, middleware.checkReviewExistence,
function (req, res) {
Campground.findById(req.params.id, function (err, campground) {
if (err) {
req.flash("error", err.message);
return res.redirect("back");}
res.render("reviews/new", {campground: campground});
});});
// Reviews Create
router.post("/",    middleware.isLoggedIn,    middleware.checkReviewExistence,
function (req, res) {
```

```
Campground.findById(req.params.id).populate("reviews").exec(function (err,
campground) {
if (err) {
req.flash("error", err.message);
return res.redirect("back");}
Review.create(req.body.review, function (err, review) {
if (err) { req.flash("error", err.message);
return res.redirect("back");}
review.author.id = req.user._id;
review.author.username = req.user.username;
review.campground = campground;
review.save();
campground.reviews.push(review);
campground.rating = calculateAverage(campground.reviews);
campground.save();
req.flash("success", "Your review has been successfully added.");
res.redirect('/campgrounds/' + campground._id);
});});});
// Reviews Edit
router.get("/:review_id/edit", middleware.checkReviewOwnership, function
(req, res) {
Review.findById(req.params.review_id, function (err, foundReview) {
if (err) {req.flash("error", err.message);
return res.redirect("back"); }
res.render("reviews/edit", {campground_id: req.params.id, review:
foundReview});});});
// Reviews Update
```

```
router.put("/:review_id", middleware.checkReviewOwnership, function (req,
res) {
Review.findByIdAndUpdate(req.params.review_id, req.body.review, {new:
true}, function (err, updatedReview) {
if (err) {
req.flash("error", err.message);
return res.redirect("back"); }
Campground.findById(req.params.id).populate("reviews").exec(function (err,
campground) {
if (err) {
req.flash("error", err.message);
return res.redirect("back"); }
campground.rating = calculateAverage(campground.reviews);
campground.save();
req.flash("success", "Your review was successfully edited.");
res.redirect('/campgrounds/' + campground._id);});});});
// Reviews Delete
router.delete("/:review_id", middleware.checkReviewOwnership, function (req,
res) {Review.findByIdAndRemove(req.params.review_id, function (err) {
if (err) {
req.flash("error", err.message);
return res.redirect("back");  }
Campground.findByIdAndUpdate(req.params.id, {$pull: {reviews:
req.params.review_id}}, {new: true}).populate("reviews").exec(function (err,
campground) {
if (err) {
req.flash("error", err.message);
return res.redirect("back");}
campground.rating = calculateAverage(campground.reviews);
campground.save();
req.flash("success", "Your review was deleted successfully.");
res.redirect("/campgrounds/" + req.params.id);});});});
function calculateAverage(reviews) {
if (reviews.length === 0) {
return 0;}
var sum = 0;
reviews.forEach(function (element) {
sum += element.rating;});
return sum / reviews.length;}
module.exports = router;
```

5.6 Module Campgrounds

This Module contains the html code for different places.

5.6.1 Edit.ejs

This file contains the html code for editing places.

```
<% include ../partials/header %>
<div class="row">
<h1 style="text-align: center">Edit Your Place <h3 style="text-align: center"><%= campground.name%></h3></h1>
<div style="width: 30%; margin: 25px auto">
<form action="/campgrounds/<%= campground._id %>?_method=PUT" method="POST">
<div class="form-group">
<input class="form-control" type="text" name="campground[name]" value="<%= campground.name%>">
</div>
<div class="form-group">
<input class="form-control" type="number" name="campground[price]" value="<%= campground.price %>" min="0.00" step="0.01">
 </div>
<div class="form-group">
<input class="form-control" type="text" name="campground[image]" value="<%= campground.image%>">
</div>
```

```
<div class="form-group">
<input class="form-control" type="text" name="campground[description]" value="<%= campground.description%>">
</div>
<div class="form-group">
<input class="form-control" type="text" name="campground[youtubeurl]" value="<%= campground.youtubeurl%>">
</div><div class="form-group">
<button class="btn btn-lg btn-primary btn-block">Submit!</button>
</div></form>
<a href="/campgrounds">Go Back</a> </div></div>
<% include ../partials/footer %>
```

5.6.2 Index.ejs

This file contains the html code for different places.

```
<% include ../partials/header %><script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/1.6.0/jquery.min.js"></script>
<script type="text/javascript">
$(function(){
$('#button').click(function(){
if(!$('#iframe').length) {
$('#iframeHolder').html('<iframe id="iframe" src="https://console.dialogflow.com/api-client/demo/embedded/bc0f67d3-4fe9-
4d3e-a081-11ca4f6bf6e8" width="250" height="300"
style="float:right"></iframe>'); }}}); });
</script>
<button class="btn btn-md btn-info" id="button" style="float:right"><i
class="fas fa-question fa-1px"></i>Ask Questions</button>
<div id="iframeHolder"></div>
<a href="https://www.facebook.com/" class="btn btn-md btn-info "
role="button" style="float:right;"><i class="fab fa-facebook-square
fa-lg"></i>FaceBook</a>
<a href="https://www.instagram.com/accounts/login/?hl=en" class="btn btn-md
btn-info " role="button" style="float:right;" ><i class="fab fa-instagram
fa-lg"></i>Insta</a>
<a href="https://www.youtube.com/" class="btn btn-md btn-info "
role="button" style="float:right;"><i class="fab fa-youtube
fa-lg"></i>YouTube</a>
<header  class="jumbotron " >
<div class="container" >
<h1 style="color:black">Welcome To TouriSum!</h1>
<p style="color:black  ; font-size:25px">Only website where you can share
your experience about visited places</p>
<p> <a class="btn btn-primary btn-lg" href="/campgrounds/new">
Add your own place</a></p></div> </header>
<div class="row text-center" style="display:flex; flex-wrap:wrap; " >
<div class="col-md-12">
<% if( noMatch !== null) { %>
<h3><%= noMatch %></h3>
<p>
```

```
<form action="/campgrounds" method="GET" class="form-inline">
<div class="form-group">
<input    type="text"    name="search"    placeholder="Search    places..."
class="form-control">
<input type="submit" value="Search" class="btn btn-default">
</div></form></p> <%  }  %></div>
<% campgrounds.forEach(function(campground){ %>
<div class="col-md-3 col-sm-6">
<div class="thumbnail">
<img src="<%= campground.image %>">
<div class="caption"><h4><%= campground.name %></h4>
<% if (campground.rating === 0) { %>
<em>No reviews yet.</em><% } else { %>
<span class="fa fa-star checked"></span>
<span class="fa fa-star <% if (campground.rating > 1.5) { %> checked <% }
%>"></span>
<span class="fa fa-star <% if (campground.rating > 2.5) { %> checked <% }
%>"></span>
<span class="fa fa-star <% if (campground.rating > 3.5) { %> checked <% }
%>"></span>
<span class="fa fa-star <% if (campground.rating > 4.5) { %> checked <% }
%>"></span>
<%  }  %></div><a  href="/campgrounds/<%=  campground._id%>"  class="btn
btn-primary"> More Info</a></p>
</div></div><% }) %></div>
<div class="row text-center">
<% if (pages && pages > 0) { %>
```

```
<ul class="pagination text-center">
<% if (current == 1) { %>
<li class="disabled"><a>First</a></li>
<% } else { %>
<li><a
href="/campgrounds<%if(search){%>?search=<%=search%><%}%>">First</a
></li> <% } %>
<% if (current == 1) { %>
<li class="disabled"><a>«</a></li>
<% } else { %>
<li><a         href="/campgrounds?page=<%=         Number(current)         -         1
%><%if(search){%>&search=<%=search%><%}%>">«</a></li> <% } %>
<% var i = (Number(current) > 5 ? Number(current) - 4 : 1) %>
<% if (i !== 1) { %><li class="disabled"><a>...</a></li>
<% } %> <% for (; i <= (Number(current) + 4) && i <= pages; i++) { %>
<% if (i == current) { %><li class="active"><a><%= i %></a></li>
<%     }     else     {     %>     <li><a     href="/campgrounds?page=<%=     i
%><%if(search){%>&search=<%=search%><%}%>"><%= i %></a></li><%
}   %>   <%   if   (i   ==   Number(current)   +   4   &&   i   <   pages)   {   %><li
class="disabled"><a>...</a></li> <% } %> <% } %>
<% if (current == pages) { %>
<li class="disabled"><a>»</a></li>
<% } else { %>
<li><a         href="/campgrounds?page=<%=         Number(current)         +         1
%><%if(search){%>&search=<%=search%><%}%>">»</a></li> <% } %>
<% if (current == pages) { %>
<li class="disabled"><a>Last</a></li>
<% } else { %>
<li><a href="/campgrounds?page=<%= pages %><%if(search){%>&search=<%=search%><%}%>">Last</a></li>
<% } %></ul><% } %></div>
<% include ../partials/footer %>
```

5.6.3 New.ejs

This file contains the html code for adding new places.

```
<% include ../partials/header %>
<div class="row">
<h1 style="text-align: center"> Add your place here!</h1>
<div style="width: 30%; margin: 25px auto">
<form action="/campgrounds" method="POST">
<div class="form-group">
<input class="form-control" type="text" name="name" placeholder="enter place name"> </div>
<div class="form-group">
<input class="form-control" type="number" name="price" placeholder="enter entry fee " min="0.00" step="0.01"></div>
 <div class="form-group">
<input class="form-control" type="text" name="image" placeholder="enter place img url"></div>
```

```
<div class="form-group"><input class="form-control" type="text"
name="description" placeholder="description of place"> </div>
<div class="form-group">
<input class="form-control" type="text" name="youtubeurl"
placeholder="youtube reviews link of place"> </div>
<div class="form-group">
<button class="btn btn-lg btn-primary btn-block">Submit!</button>
</div>   </form>
<a href="/campgrounds">Go Back</a></div></div>
<% include ../partials/footer %>
```

5.6.4 Show.ejs

This file contains the html code for More Info page.

```
<% include ../partials/header %>
<div class="row">
<div class="col-md-3">
<p class="lead">TouriSum</p>
<div class="list-group">
<li class="list-group-item active  text-center"> Place Info</li></div>
<div><a title="click to view map" target="_blank"
href="https://www.google.com/maps/d/u/0/viewer?f=q&hl=en&geocode&ie=U
TF8&msa=0&ll=24.88394507726512%2C74.63527700000009&spn=0.008738
%2C0.019248&source=embed&mid=1TLtkEy0HBOOgyPy_p2OUzxJpG24&z
=16" >
<div class="cmap"><img
```

```
id="map"
src='https://thumbs.gfycat.com/SaneValidBaleenwhale-size_restricted.gif'>
</div></a> </div> </div>
<div class="col-md-9">
<div class="thumbnail">
<a href="#"> <img src="<%= campground.image%>"></a>
<div class="caption-full">
<h4 class="pull-right">Entry fee ₹<%= campground.price%></h4>
<h4><a><%= campground.name%></a></h4
<p><%= campground.description%></p>
<p>
<em>Place added by:<a href="/users/<%= campground.author.id
%>"><%=campground.author.username %></a>, <%=
moment(campground.createdAt).fromNow() %></em></p>
<a href="<%=campground.youtubeurl %>">Click this link to view youtube
reviews of this place </a><br>
<% if( currentUser && campground.author.id.equals(currentUser._id)){ %>
<a class="btn btn-warning"
href="/campgrounds/<%=campground._id%>/edit">Edit Place</a
<form class="delete-form"
action="/campgrounds/<%=campground._id%>?_method=DELETE"
method="POST">
<button class="btn btn-danger">Delete Place</button>
</form><% } %></div> </div>
<!--Review section→
<div class="thumbnail">
<div class="caption-full">
```

```
<% if (campground.rating === 0) { %>
<h5><em>No reviews yet.</em></h5><% } else { %><p>
<span class="fa fa-star checked"></span>
<span class="fa fa-star <% if (campground.rating > 1.5) { %> checked <% }
%>"></span>
<span class="fa fa-star <% if (campground.rating > 2.5) { %> checked <% }
%>"></span>
<span class="fa fa-star <% if (campground.rating > 3.5) { %> checked <% }
%>"></span>
<span class="fa fa-star <% if (campground.rating > 4.5) { %> checked <% }
%>"></span>
<em>(total reviews: <%= campground.reviews.length %>)</em>
</p><p>Current campground rating: <strong><%=
campground.rating.toFixed(2) %></strong></p>
<p><h4>Latest reviews for this campground:</h4></p>
<hr style="margin-top: 0;">
<% campground.reviews.slice(0, 5).forEach(function(review){ %>
<div class="row">
<div class="col-md-3">
<%- '<span class="fa fa-star checked"></span>'.repeat(review.rating) %><%-
'<span class="fa fa-star"></span>'.repeat(5 - review.rating) %>
<div>Review by: <strong><%= review.author.username %></strong></div>
<span><em><%= review.updatedAt.toDateString() %></em></span> </div>
<div class="col-md-9">
<p style="text-align: justify; word-wrap: break-word;">
<%= review.text %></p>
<% if(currentUser && review.author.id.equals(currentUser._id)){ %>
<a class="btn btn-xs btn-warning" href="/campgrounds/<%=campground._id
%>/reviews/<%=review._id %>/edit">Edit</a>
<form id="delete-form" action="/campgrounds/<%=campground._id
%>/reviews/<%=review._id %>?_method=DELETE" method="POST">
<input type="submit" class="btn btn-xs btn-danger" value="Delete">
</form><% } %></div></div><hr><% }); %>
<div style="margin-bottom: 10px;">
<h4><a href="/campgrounds/<%= campground._id %>/reviews"><i class="fa
fa-search" aria-hidden="true"></i> See all reviews</a></h4></div> <% }
%><div><a class="btn btn-primary btn-lg <% if (currentUser &&
campground.reviews.some(function (review) {return
review.author.id.equals(currentUser._id)})) { %> disabled <% } %>"
href="/campgrounds/<%= campground._id %>/reviews/new">
Write a New Review</a></div></div></div>
<!--comments section→
<div class="well">
<div class="text-right"><a class="btn btn-success"
href="/campgrounds/<%=campground._id%>/comments/new">Add New
Comment</a> </div> <hr>
<% campground.comments.forEach(function(comment){ %>
<div class="row">
<div class="col-md-12">
<strong><%= comment.author.username %></strong>
<span class="pull-right"><%= moment(comment.createdAt).fromNow()
%></span><p> <%= comment.text %></p><% if( currentUser &&
comment.author.id.equals(currentUser._id)){ %>
<a class="btn btn-xs btn-warning"
```

```
href="/campgrounds/<%=campground._id%>/comments/<%=comment._id%>/
edit">Edit</a>
<form                                                        class="delete-form"
action="/campgrounds/<%=campground._id%>/comments/<%=comment._id%
>?_method=DELETE" method="POST">
<input type="submit" class="btn btn-xs btn-danger" value="Delete">
</form><% } %></div></div><% }) %></div></div> </div>
<% include ../partials/footer %>
```

5.7 Module Partials

This module contains the code for the header and footer part of every page on the web app.

5.7.1 Header.ejs

This file contains the html code and different types of cdn used in our web app.

```
<!DOCTYPE html>
<html> <head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>TouriSum</title>
<linkrel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css"
integrity="sha384-HSMxcRTRxnN+Bdg0JdbxYKrThecOKuH5zCYotlSAcp1+
c8xmyTe9GYg1l9a69psu" crossorigin="anonymous">
<link rel="stylesheet" href="/stylesheets/main.css">
<linkrel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.11.2/css/all.css">
</head><body>
<nav class="navbar navbar-default">
<div class="container">
<div class="navbar-header"><button type="button" class="navbar-toggle
collapsed" data-toggle="collapse" data-target="#navbar" aria-expanded="false"
aria-controls="navbar">
<span class="sr-only">Toggle navigation</span>
<span class="icon-bar"></span>
```

```
<span class="icon-bar"></span>
<span class="icon-bar"></span>
 </button>
<a class="navbar-brand" href="/">TouriSum</a></div>
<div id="navbar" class="collapse navbar-collapse">
<ul class="nav navbar-nav">
<li class="<%= typeof page !== 'undefined' && page === 'campgrounds' ?
'active' : '' %>"><a href="/campgrounds"><i class="fas fa-home
fa-lg"></i>Home</a>  </li></ul>
<ul class="nav navbar-nav" ><li>
<form style="margin-top:8px" action="/campgrounds" method="GET"
class="form-inline">
<div class="form-group">
<input type="text" name="search" placeholder="Search place..."
class="form-control">
<input type="submit" value="Search" class="btn btn-default">
</div></form></li></ul>
<ul class="nav navbar-nav navbar-right">
<% if(!currentUser){ %>
<li class="<%= typeof page !== 'undefined' && page === 'login' ? 'active' : ''
%>"><a href="/login">Login</a></li>
<li class="<%= typeof page !== 'undefined' && page === 'register' ? 'active' : ''
%>"><a href="/register">Sign Up</a></li>
<% } else { %>
<li><a href="/users/<%= currentUser._id %>">Signed In As <%=
currentUser.username %></a></li>
<li><a href="/logout">Logout</a></li>
<% } %></ul></div></div></nav>
<div class="container">
<% if(error && error.length > 0){%> <div class="alert alert-danger"
role="alert">
<%= error %> </div><% }%>
<% if(success && success.length > 0){%>
<div class="alert alert-success" role="alert">
<%= success %> </div> <% }%></div><div class="container">
```

5.7.2 Footer.ejs

This file contains the html code and javascript cdn for this web app.

```
</div><script src="https://code.jquery.com/jquery-3.1.1.min.js"
integrity="sha256-hVVnYaiADRTO2PzUGmuLJr8BLUSjGIZsDYGmIJLv2b8
=" crossorigin="anonymous"></script>
<!-- Bootstrap JS CDN -->
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2mCWN
IpG9mGCD8wGNIcPD7Txa" crossorigin="anonymous"></script>
</body></html>
```

6. CONCLUSION

I have successfully completed the project under the guidance of “Mr.Subhash Chandra Gupta”, he helped me at every point where I needed him. This project “Tourism Web Application” not only helps users to take information about places like location, fare, nearby hotels, railway, airports etc but it also provides features like adding the photos of that place, rate that place, do comments, share their experience and much more. To build this app I have used technologies and languages like html, css, bootstrap, javascript, nodeJs, express, mongoose, mongoDB, git, heroku. This project can help tourists from all over the world who are going to visit some place as a tourist. Before going to that place they can gather all the information about their journey and experience of the others who already visited that place. Our website has a very easy to use user interface, it is scalable and can handle thousands of users at once.

In future this web application can be scaled to a large extent and can be changed and updated for more functionalities and features according to the users needs.

7. REFERENCES

1. https://www.udemy.com/course/the-web-developer-bootcamp/
2. https://www.youtube.com/
3. www.google.com
4. www.stackoverflow.com
5. www.w3schools.com
6. www.wikipedia.com