PROJECT REPORT

On

YELPCAMP

Submitted as partial fulfilment for the award of

MASTER OF COMPUTER APPLICATION DEGREE

SESSION 2021- 2022 By

SAUMYA SHUKLA

2000320140116

Under the guidance of

Prof. (Dr.) Devendra Kumar
HOD-MCA

Mr. RAKESH KUMAR
S/W ENGINEER

ABES ENGINEERING COLLEGE (032)

AFFILIATED TO Dr. A.P.J ABDUL KALAM TECHNICAL UNIVERSITY, UTTAR PRADESH LUCKNOW

WEBKUL

Webkul Software Pvt. Ltd.
CIN: U72300UP2010PTC041790
Address: H-28 ARV Park
Sector 63 Noida 201301 (UP) India
Phone: +91 9870284067
Website: webkul.com

24th May 2022

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Ms. **Saumya Shukla** D/o Mr. **Suneet Shukla** , a student of **MCA-4th Semester (Roll. No. 2000320140087)**, Dr. APJ Abdul Kamal University, Lucknow has successfully completed her internship under the guidance of **Mr. Sanjeev Papnoi (Sr. Software Engineer)** bearing project title "**Yelpcamp** from 15th Feb 2022 to 15th May 2022 at **Webkul software Private Limited**

.

During the period of her internship with us she was found punctual, hardworking and Inquisitive.

We wish her all the best for her upcoming future.

Regards,

Jennifer Bennett
HR Executive

STUDENT DECLARATION

I hereby declare that the work being presented in this report entitled **“YELPCAMP”** is an authentic record of my own work carried out under the supervision of **Prof. (Dr.) Devendra Kumar, HOD-MCA.**

The matter embodied in this report has not been submitted by me for the award of any other degree.

Date: 15-05-2022

Signature of Student

SAUMYA SHUKLA

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Signature of HOD

Prof. (Dr.) Devendra Kumar

HOD-MCA

Signature of Internal

Prof. (Dr.) Devendra Kumar

HOD-MCA

Signature of External

Mr. Rakesh Kumar

S/W Engineer

ACKNOWLEDGEMENT

I express my sincere gratitude to all those who initiated and helped me in the successful completion of my project. One of the most pleasing aspect in collecting the necessary information and compiling it is the opportunity to thanks those who have actively contributed to it.

We would like to express thanks to **Prof.** (**Dr.) Devendra Kumar (HOD - MCA)** for his guidance and cooperation render for following our to undergo training under his guidance we're also thanks to all the other staff of college who had helped me in spite of their busy schedule.

I am highly indebted to **Prof.** (**Dr.) Devendra Kumar (HOD - MCA)** for his continuous effort in building a good infrastructure and develop a professional attitude within ourselves during the academic period of MCA IVth sem.

Signature of Student

SAUMYA SHUKLA

2000320140087

MCA IVth semester

ABSTRACT

The YelpCamp is a project focused on providing people a platform where they can add the pictures, price per night, review of the places they have visited before. It helps in the decision making of the people after we have a good number of reviews to understand if this place is worth the money or not. It can be thought of as an IMDB for the campgrounds.

TABLE OF CONTENT

LIST OF FIGURE

1. INTRODUCTION

This project is designed for use by people to rate and review the places they have visited exactly like we have IMDB for movie ratings and reviews.

It allows visitors to rate the places out of five stars they have visited while they can also write a review on the website. With the help of inbuilt world map it also makes it easy to locate the place on the map.

It allows users to create accounts to post pictures of the campgrounds.

1.1 OBJECTIVE

The main objectives of our project are:

Platform : To provide users a platform where they checkout places and read their reviews of make decision of visiting the place.

Awareness: To make people aware of their surrounding campgrounds with per night charges and how good or bad they are according to the ratings and reviews provided by the previous visitors and account holders.

Experience: To allow people to share experiences of their visit to a particular place which can turn out to be a deciding factor for future visitors to that place.

1.2 NEED OF PROJECT

The need of the project arises from the unavailability of such platform where users can rate and review campgrounds and can also add new as they visit more. Other than that users without creating an account just look at each campground's rating and review to make decision to visit that place if they find it worth the price. This project is developed keeping in mind the requirements of these users. To offer an interactive platform with an interactive map in the platform itself to give user a pleasing and hassle-free experience.

2. FEASIBILITY STUDY

2.1 PROPOSED SYSTEM

The features and the benefits of the proposed system are : -

1.It includes a registration and login portal for the users.

2. No information is repeated anywhere in memory. This will ensure economical use of storage space and consistency of stored data.

3. Inbuilt responsive world map on the website which allows users to pinpoint the location and provides a good visual experience to the user.

4. This saves a lot of time and money.

5. It makes people aware of the campgrounds which they have not visited but they are planning to visit beforehand.

2.2 CURRENT SCOPE

Deliverables :

The deliverables of this project will be the application hosted on a web server and the bundled codebase in a zip file.

Features:

1. Web Portal: Web application to provide a user interface to the user to perform all the functions provided by the application.

2. Campgrounds Images: User will be able to add campground images while adding a campground to the application.

3. Inbuilt Map: User will be able to locate the place of the campground using the inbuilt map in the application.

4. Authentication: Users will be able to create accounts and login to add a campground. No user who is not logged in will be able to add a campground or rate any of the already added campgrounds.

3. SOFTWARE SPECIFIC REQUIREMENT

3.1 SPECIFIC REQUIREMENTS

3.1.1 FUNCTIONAL REQUIREMENTS

Functional requirements: are requirements used to illustrate the inner workings of the system, describe the system, and explain each subsystem.
It includes the tasks that the system must perform, the processes involved, the data and system must contain, and the interfaces with the users. The functional requirements of this project are:

- Display all the campgrounds to all the visitors on the website.
- Only users who are logged in should be able to create new campgrounds.
- Session should be maintained for the user who is logged in.
- Web application must contain a world map to pinpoint the location of the campground.
- Users should be able to rate from 1 to 5 stars and write a review.
- Any other user should not be able to change other user's rating or review on a campground.
- Users should be able to upload pictures of the campgrounds.
- Users must mention the per night charges and location of the campground.

Login and Signup functionality in the application.

3.1.2 NON-FUNCTIONAL REQUIREMENTS

Non-Functional Requirements**:** Describe aspects of the system related to how the system delivers the functional requirements. That is:

- Security
- Availability
- Ease of use
- Error Handling

4. TECHNOLOGIES USED

4.1 FRONTEND

Html5

HTML5 is the latest and most enhanced version of HTML. Technically, HTML is not a programming language, but rather a markup language. In this tutorial, we will discuss the features of HTML5 and how to use it in practice

EJS

What is the "E" for? "Embedded?" Could be. How about "Effective," "Elegant," or just "Easy"? EJS is a simple templating language that lets you generate HTML markup with plain JavaScript. No religiousness about how to organize things. No reinvention of iteration and control-flow. It's just plain JavaScript.

JavaScript

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

4.2 BACKEND

Node Js

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Express js

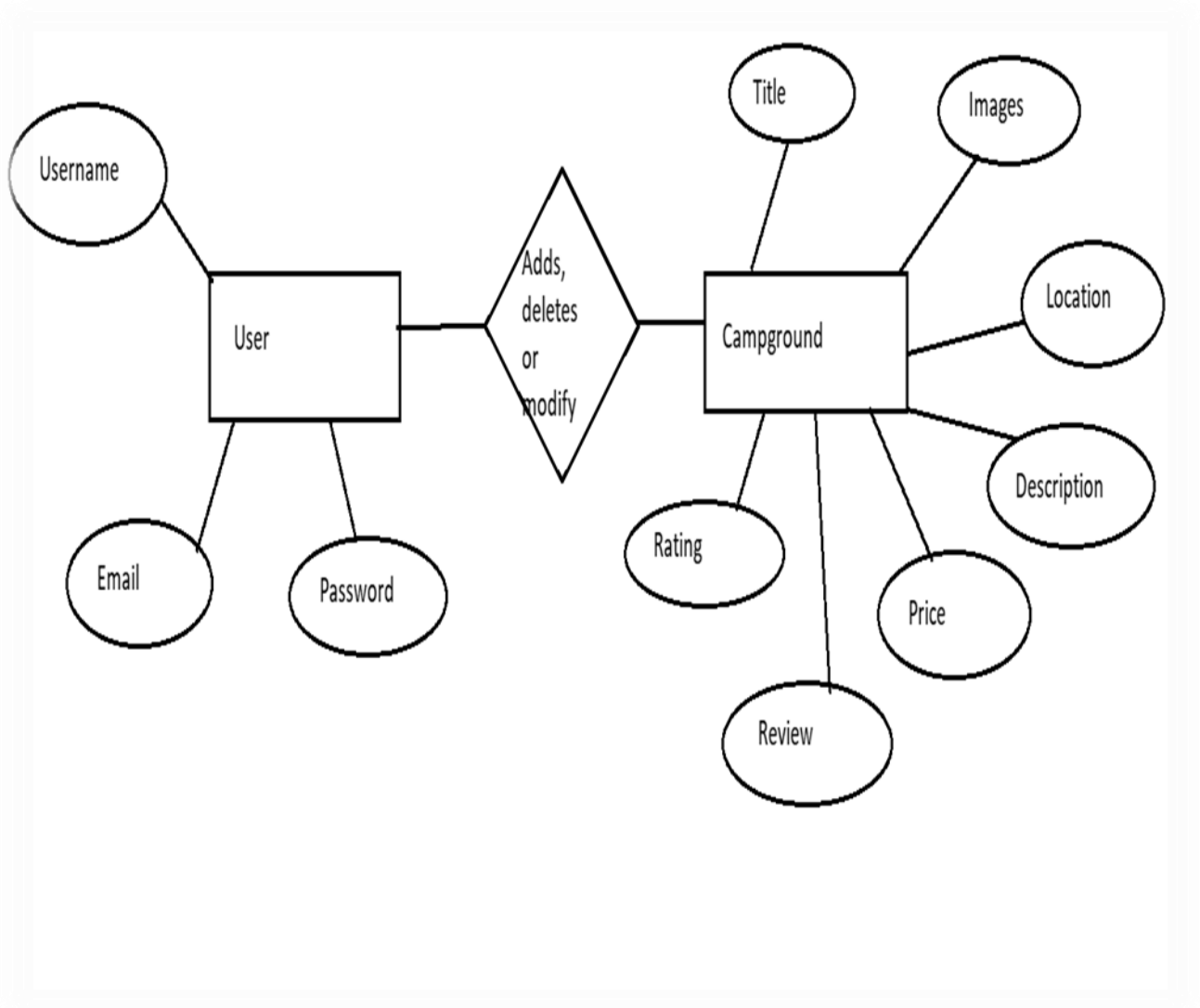
Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

Mongo DB

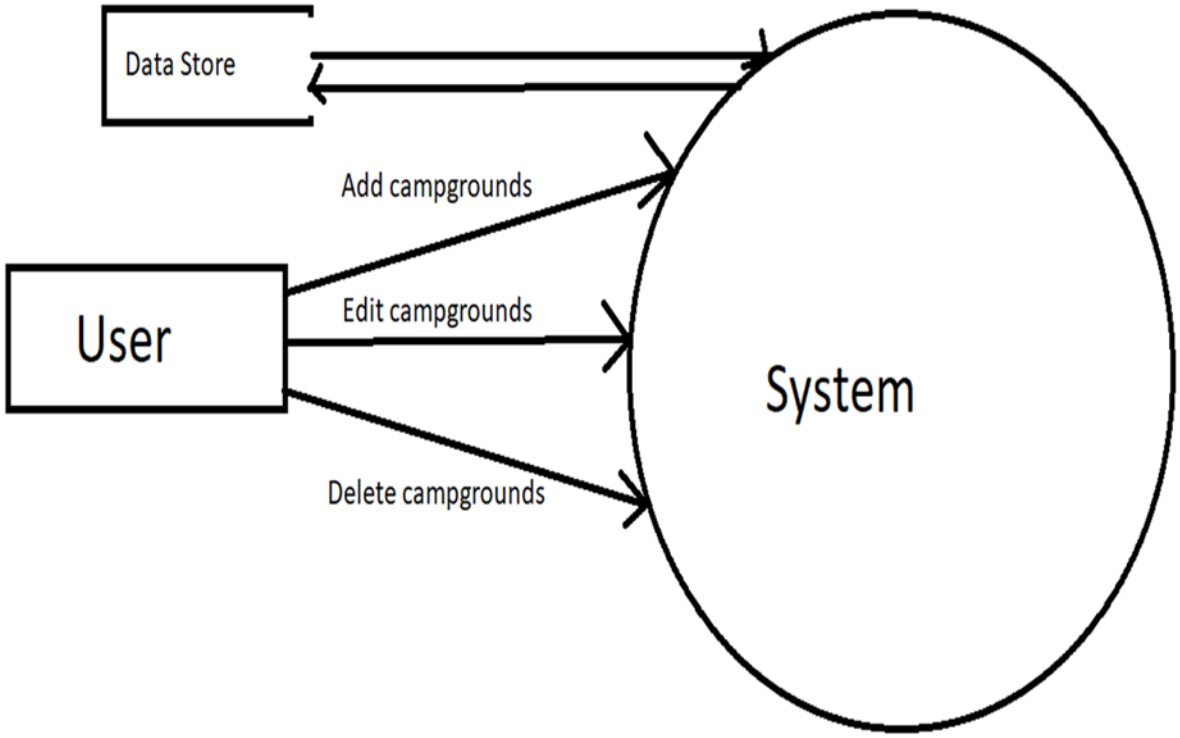
MongoDB is an open-source document database and leading NoSQL database. MongoDB is written in C++. This tutorial will give you great understanding on MongoDB concepts needed to create and deploy a highly scalable and performance-oriented database.

5. DESIGN

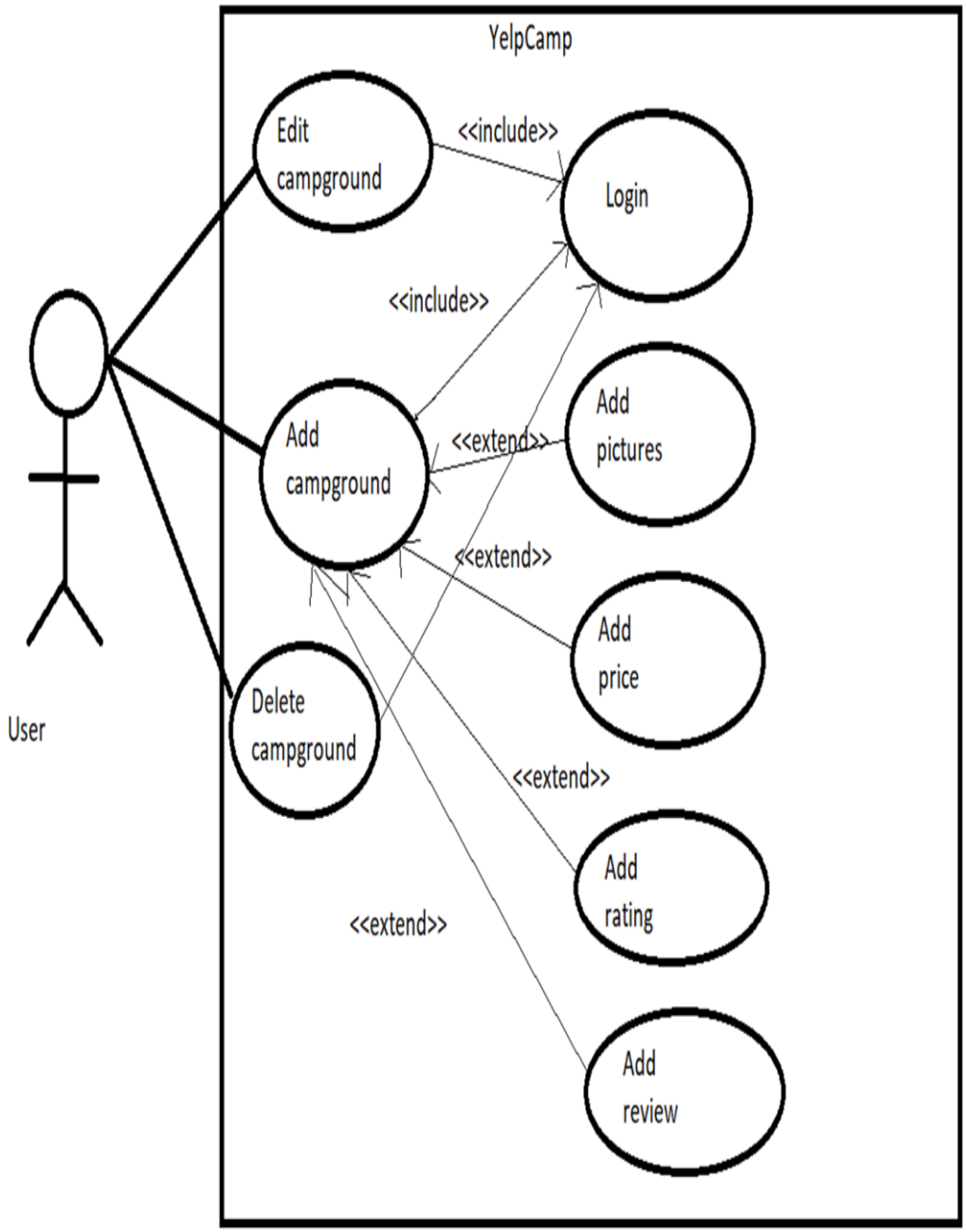
5.1 ER-Diagram



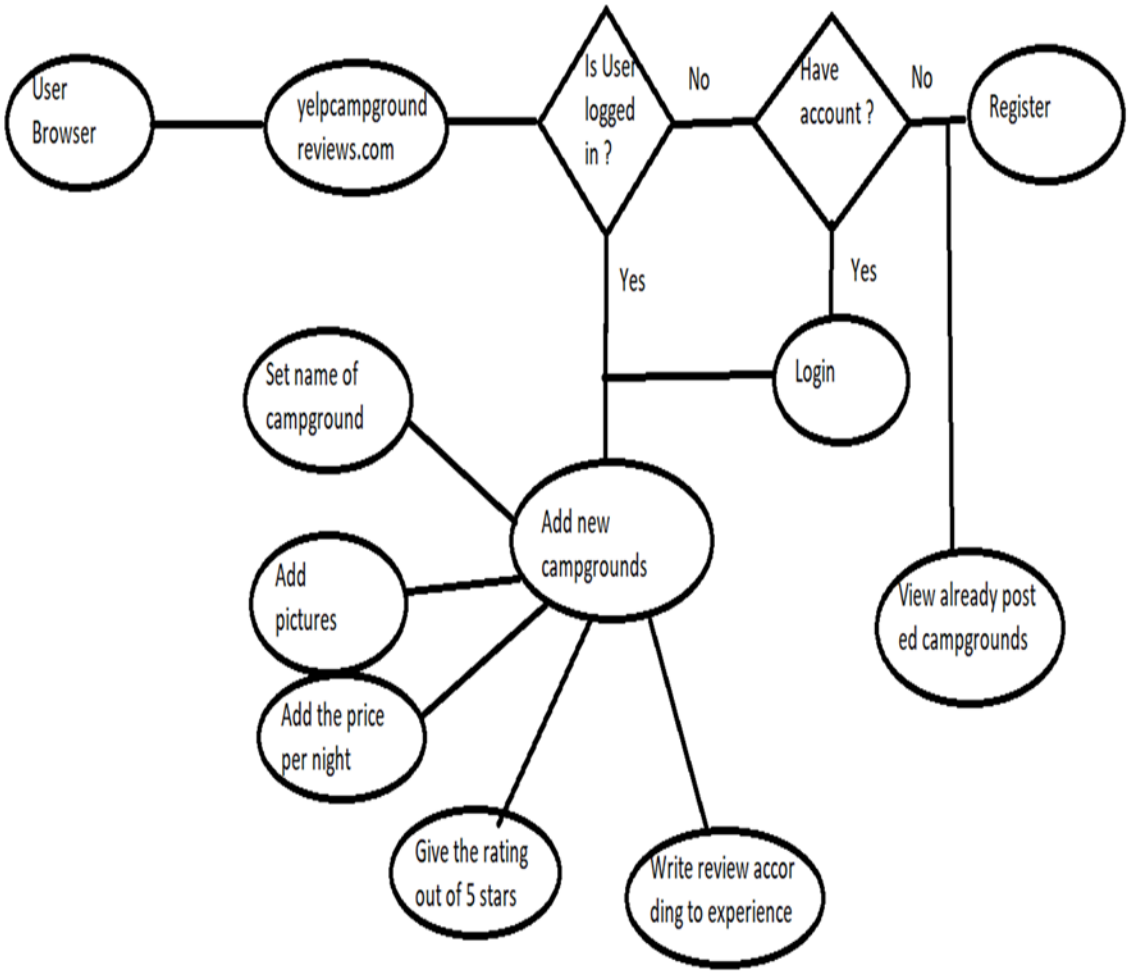
5.2 DATA FLOW DIAGRAM (0-level DFD)



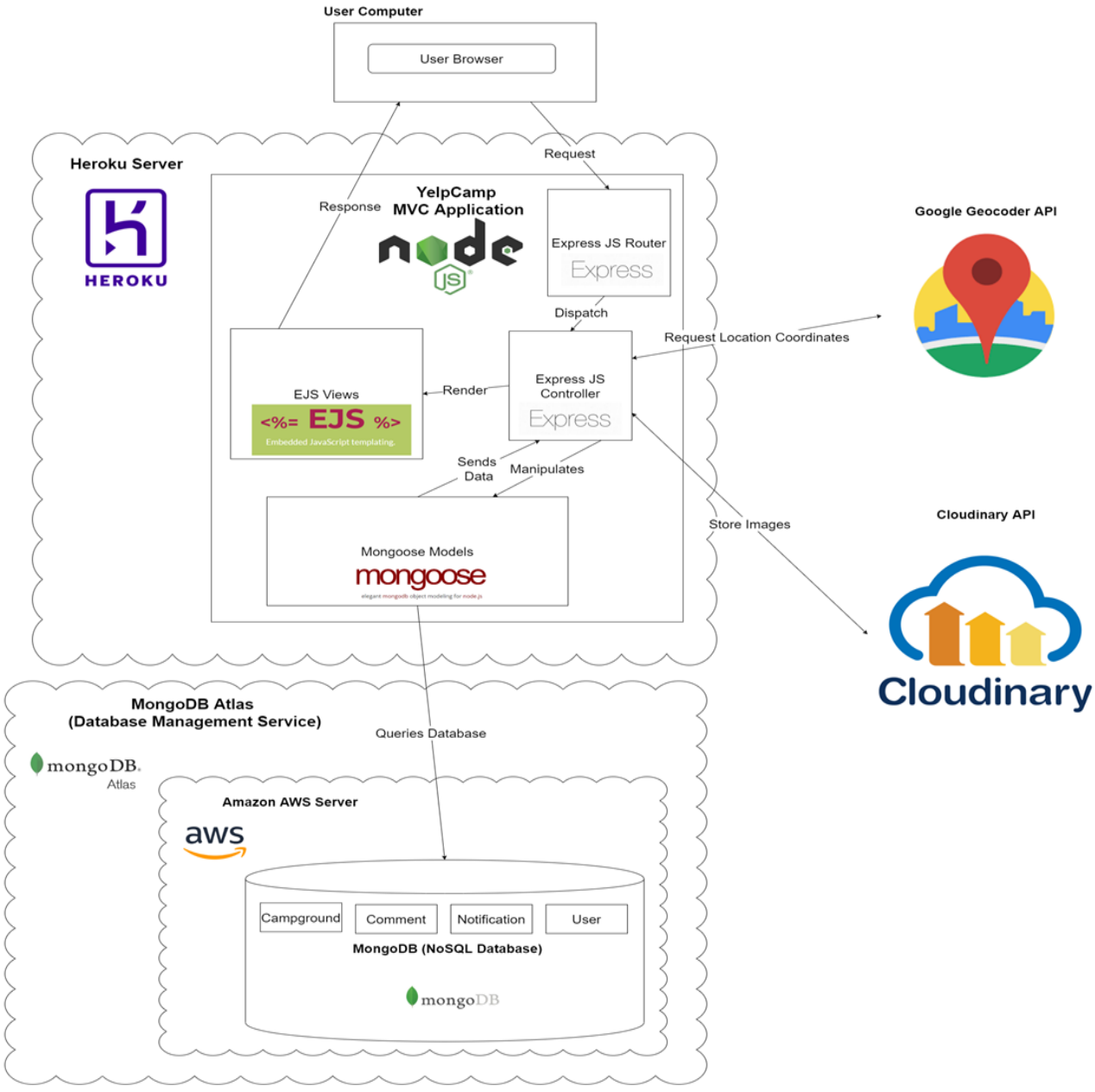
5.3 USE CASE DIAGRAM



5.4 ACTIVITY DIAGRAM



5.5 DEPLOYMENT DIAGRAM



6. DETAILS OF MODULES

User Authentication:

User Authentication module provide users to create their account on the website and then they can login and logout according to their need.

Models:

Models take are of all the database and schema related work and also the ORM (Object Relational Mapping).

Views:

Views takes care of all the views that will be available to the users. User authorization takes care of who is authorized to view a certain page.

Controller:

Controller binds the models and views together. It take input data from the views and make changes to the model according to that. It makes views and models to be interdependent on each other.

7. CODING

App.js

```
const express = require('express');
const path = require('path');
const mongoose = require('mongoose');
const ejsMate = require('ejs-mate');
const session = require('express-session');
const flash = require('connect-flash');
const ExpressError = require('./utils/ExpressError');
const methodOverride = require('method-override');
const passport = require('passport');
const LocalStrategy = require('passport-local');
const User = require('./models/user');
const helmet = require('helmet');
const mongoSanitize = require('express-mongo-sanitize');
const userRoutes = require('./routes/users');
const campgroundRoutes = require('./routes/campgrounds');
const reviewRoutes = require('./routes/reviews');

const MongoDBStore = require("connect-mongo")(session);

const dbUrl = process.env.DB_URL || 'mongodb://localhost:27017/yelp-camp';

mongoose.connect(dbUrl, {
    useNewUrlParser: true,
    useCreateIndex: true,
    useUnifiedTopology: true,
```

```
    useFindAndModify: false
});

const db = mongoose.connection;
db.on("error", console.error.bind(console, "connection error:"));
db.once("open", () => {
    console.log("Database connected");
});

const app = express();

app.engine('ejs', ejsMate)
app.set('view engine', 'ejs');
app.set('views', path.join(__dirname, 'views'))

app.use(express.urlencoded({ extended: true }));
app.use(methodOverride('_method'));
app.use(express.static(path.join(__dirname, 'public')))
app.use(mongoSanitize({
    replaceWith: '_'
}))
const secret = process.env.SECRET || 'thisshouldbeabettersecret!';

const store = new MongoDBStore({
    url: dbUrl,
    secret,
    touchAfter: 24 * 60 * 60
});

store.on("error", function (e) {
```

```
    console.log("SESSION STORE ERROR", e)
})

const sessionConfig = {
    store,
    name: 'session',
    secret,
    resave: false,
    saveUninitialized: true,
    cookie: {
        httpOnly: true,
        // secure: true,
        expires: Date.now() + 1000 * 60 * 60 * 24 * 7,
        maxAge: 1000 * 60 * 60 * 24 * 7
    }
}

app.use(session(sessionConfig));
app.use(flash());
app.use(helmet());


const scriptSrcUrls = [
    "https://stackpath.bootstrapcdn.com",
    "https://api.tiles.mapbox.com",
    "https://api.mapbox.com",
    "https://kit.fontawesome.com",
    "https://cdnjs.cloudflare.com",
    "https://cdn.jsdelivr.net",
];
```

```
const styleSrcUrls = [
    "https://kit-free.fontawesome.com",
    "https://stackpath.bootstrapcdn.com",
    "https://api.mapbox.com",
    "https://api.tiles.mapbox.com",
    "https://fonts.googleapis.com",
    "https://use.fontawesome.com",
];
const connectSrcUrls = [
    "https://api.mapbox.com",
    "https://*.tiles.mapbox.com",
    "https://events.mapbox.com",
];
const fontSrcUrls = [];
app.use(
    helmet.contentSecurityPolicy({
        directives: {
            defaultSrc: [],
            connectSrc: ["'self'", ...connectSrcUrls],
            scriptSrc: ["'unsafe-inline'", "'self'", ...scriptSrcUrls],
            styleSrc: ["'self'", "'unsafe-inline'", ...styleSrcUrls],
            workerSrc: ["'self'", "blob:"],
            childSrc: ["blob:"],
            objectSrc: [],
            imgSrc: [
                "'self'",
                "blob:",
                "data:",
                "https://res.cloudinary.com/douqbebwk/", //SHOULD MATCH YOUR
CLOUDINARY ACCOUNT!
```

```
                "https://images.unsplash.com",
            ],
            fontSrc: ["'self'", ...fontSrcUrls],
        },
    })
);


app.use(passport.initialize());
app.use(passport.session());
passport.use(new LocalStrategy(User.authenticate()));

passport.serializeUser(User.serializeUser());
passport.deserializeUser(User.deserializeUser());

app.use((req, res, next) => {
    res.locals.currentUser = req.user;
    res.locals.success = req.flash('success');
    res.locals.error = req.flash('error');
    next();
})


app.use('/', userRoutes);
app.use('/campgrounds', campgroundRoutes)
app.use('/campgrounds/:id/reviews', reviewRoutes)


app.get('/', (req, res) => {
    res.render('home')
```

```
});


app.all('*', (req, res, next) => {
    next(new ExpressError('Page Not Found', 404))
})

app.use((err, req, res, next) => {
    const { statusCode = 500 } = err;
    if (!err.message) err.message = 'Oh No, Something Went Wrong!'
    res.status(statusCode).render('error', { err })
})

const port = process.env.PORT || 3000;
app.listen(port, () => {
    console.log(`Serving on port ${port}`)
})
‘
```

Schemas.js

```
  const
  BaseJoi =
  require('joi');
                        const sanitizeHtml = require('sanitize-html');

                        const extension = (joi) => ({
                            type: 'string',
                            base: joi.string(),
                            messages: {
                                'string.escapeHTML': '{{#label}} must not include HTML!'
```

```
    },
    rules: {
        escapeHTML: {
            validate(value, helpers) {
                const clean = sanitizeHtml(value, {
                    allowedTags: [],
                    allowedAttributes: {},
                });
                if (clean !== value) return
helpers.error('string.escapeHTML', { value })
                return clean;
            }
        }
    }
});

const Joi = BaseJoi.extend(extension)

module.exports.campgroundSchema = Joi.object({
    campground: Joi.object({
        title: Joi.string().required().escapeHTML(),
        price: Joi.number().required().min(0),
        location: Joi.string().required().escapeHTML(),
        description: Joi.string().required().escapeHTML()
    }).required(),
    deleteImages: Joi.array()
});

module.exports.reviewSchema = Joi.object({
    review: Joi.object({
```

```
            rating: Joi.number().required().min(1).max(5),
            body: Joi.string().required().escapeHTML()
        }).required()
    })
```

Models/Campground.js

```
const
mongoose =
require('mon
goose');
                const Review = require('./review')
                const Schema = mongoose.Schema;


                //
                https://res.cloudinary.com/douqbebwk/image/upload/w_300/v1600113904/Y
                elpCamp/gxgle1ovzd2f3dgcpass.png

                const ImageSchema = new Schema({
                    url: String,
                    filename: String
                });

                ImageSchema.virtual('thumbnail').get(function () {
                    return this.url.replace('/upload', '/upload/w_200');
                });

                const opts = { toJSON: { virtuals: true } };

                const CampgroundSchema = new Schema({
                    title: String,
```

```
    images: [ImageSchema],
    geometry: {
        type: {
            type: String,
            enum: ['Point'],
            required: true
        },
        coordinates: {
            type: [Number],
            required: true
        }
    },
    price: Number,
    description: String,
    location: String,
    author: {
        type: Schema.Types.ObjectId,
        ref: 'User'
    },
    reviews: [
        {
            type: Schema.Types.ObjectId,
            ref: 'Review'
        }
    ]
}, opts);


CampgroundSchema.virtual('properties.popUpMarkup').get(function () {
    return `
```

```
    <strong><a href="/campgrounds/${this._id}">${this.title}</a><strong>
    <p>${this.description.substring(0, 20)}...</p>`
});



CampgroundSchema.post('findOneAndDelete', async function (doc) {
    if (doc) {
        await Review.deleteMany({
            _id: {
                $in: doc.reviews
            }
        })
    }
})

module.exports = mongoose.model('Campground', CampgroundSchema);
```

Models/user.js

```
const mongoose =
require('mongoose');
                const Schema = mongoose.Schema;
                const passportLocalMongoose = require('passport-local-mongoose');

                const UserSchema = new Schema({
                    email: {
                        type: String,
                        required: true,
```

```
        unique: true
    }
});

UserSchema.plugin(passportLocalMongoose);

module.exports = mongoose.model('User', UserSchema);
```

Models/review.js

```
const mongoose =
require('mongoose');
const Schema = mongoose.Schema;

const reviewSchema = new Schema({
    body: String,
    rating: Number,
    author: {
        type: Schema.Types.ObjectId,
        ref: 'User'
    }
});

module.exports = mongoose.model("Review", reviewSchema);
```

Views/home.ejs

```
<!DOCTY
PE html>
                <html lang="en">

                <head>
                    <meta charset="UTF-8">
                    <meta name="viewport" content="width=device-width, initial-scale=1.0">
                    <title>YelpCamp</title>
                    <link rel="stylesheet"
                href="https://stackpath.bootstrapcdn.com/bootstrap/5.0.0-
                alpha1/css/bootstrap.min.css"
                        integrity="sha384-
                r4NyP46KrjDleawBgD5tp8Y7UzmLA05oM1iAEQ17CSuDqnUK2+k9luXQ
                OfXJCJ4I" crossorigin="anonymous">
                    <link rel="stylesheet" href="/stylesheets/home.css">
                </head>

                <body class="d-flex text-center text-white bg-dark">
                    <div class="cover-container d-flex w-100 h-100 p-3 mx-auto flex-column">
                        <header class="mb-auto">
                            <div>
                                <h3 class="float-md-left mb-0">YelpCamp</h3>
                                <nav class="nav nav-masthead justify-content-center float-md-
                right">
                                    <a class="nav-link active" aria-current="page"
                href="#">Home</a>
                                    <a class="nav-link" href="/campgrounds">Campgrounds</a>
                                    <% if(!currentUser) { %>
                                    <a class="nav-link" href="/login">Login</a>
```

```
                <a class="nav-link" href="/register">Register</a>
                <% } else { %>
                <a class="nav-link" href="/logout">Logout</a>
                <% } %>
            </nav>
        </div>
    </header>
    <main class="px-3">
        <h1>YelpCamp</h1>
        <p class="lead"> Welcome to YelpCamp! <br> Jump right in and
explore our many campgrounds. <br>
            Feel free to share some of your own and comment on others!</p>
        <a href="/campgrounds" class="btn btn-lg btn-secondary font-weight-
bold border-white bg-white">View
            Campgrounds</a>
    </main>

    <footer class="mt-auto text-white-50">
        <p>&copy; 2020 </p>
    </footer>


</div>


<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
    integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxM
fooAo"
```

```
        crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/5.0.0-
alpha1/js/bootstrap.min.js"
        integrity="sha384-
oesi62hOLfzrys4LxRF63OJCXdXDipiYWBnvTl9Y9/TRlw5xlKIEHpNyvvD
Shgf/"
        crossorigin="anonymous"></script>
</body>

</html>
```

Views/partials/navbar.ejs

```
<nav
class="navbar
sticky-top
navbar-
expand-lg
navbar-dark
bg-dark">
                    <div class="container-fluid">
                        <a class="navbar-brand" href="#">YelpCamp</a>
                        <button class="navbar-toggler" type="button" data-toggle="collapse"
                    data-target="#navbarNavAltMarkup"
                            aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-
                    label="Toggle navigation">
                            <span class="navbar-toggler-icon"></span>
                        </button>
                        <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
                            <div class="navbar-nav">
```

```
                    <a class="nav-link" href="/">Home</a>
                    <a class="nav-link" href="/campgrounds">Campgrounds</a>
                    <a class="nav-link" href="/campgrounds/new">New
            Campground</a>
                </div>
                <div class="navbar-nav ml-auto">
                    <% if(!currentUser) {%>
                    <a class="nav-link" href="/login">Login</a>
                    <a class="nav-link" href="/register">Register</a>
                    <% } else {%>
                    <a class="nav-link" href="/logout">Logout</a>
                    <% } %>
                </div>
            </div>
        </nav>
```

Views/campgrounds/show.ejs

```
<link rel="stylesheet" href="/stylesheets/stars.css">
<div class="row">
    <div class="col-6">
        <div id="campgroundCarousel" class="carousel slide" data-ride="carousel">
            <div class="carousel-inner">
                <% campground.images.forEach((img, i) => {  %>
                <div class="carousel-item <%= i === 0 ? 'active' : "%>">
                    <img src="<%= img.url%>" class="d-block w-100" alt="">
                </div>
                <% }) %>
```

```
            </div>
            <% if(campground.images.length > 1) {%>
            <a class="carousel-control-prev" href="#campgroundCarousel" role="button"
data-slide="prev">
                <span class="carousel-control-prev-icon" aria-hidden="true"></span>
                <span class="sr-only">Previous</span>
            </a>
            <a class="carousel-control-next" href="#campgroundCarousel" role="button" data-
slide="next">
                <span class="carousel-control-next-icon" aria-hidden="true"></span>
                <span class="sr-only">Next</span>
            </a>
            <% } %>
        </div>

        <div class="card mb-3">
            <div class="card-body">
                <h5 class="card-title"><%= campground.title%></h5>
                <p class="card-text"><%= campground.description%></p>
            </div>
            <ul class="list-group list-group-flush">
                <li class="list-group-item text-muted"><%= campground.location%></li>
                <li class="list-group-item">Submitted by <%=
campground.author.username%></li>
                <li class="list-group-item">$<%= campground.price%>/night</li>
            </ul>
            <%  if( currentUser && campground.author.equals(currentUser._id))  {%>
            <div class="card-body">
                <a class="card-link btn btn-info"
href="/campgrounds/<%=campground._id%>/edit">Edit</a>
```

```
            <form class="d-inline"
action="/campgrounds/<%=campground._id%>?_method=DELETE" method="POST">
                <button class="btn btn-danger">Delete</button>
            </form>
        </div>
        <% } %>
        <div class="card-footer text-muted">
            2 days ago
        </div>
    </div>

</div>
<div class="col-6">
    <div id='map' ></div>

    <% if(currentUser){ %>
    <h2>Leave a Review</h2>
    <form action="/campgrounds/<%=campground._id%>/reviews" method="POST"
class="mb-3 validated-form" novalidate>
        <!-- <div class="mb-3">
            <label class="form-label" for="rating">Rating</label>
            <input class="form-range" type="range" min="1" max="5"
name="review[rating]" id="rating">
        </div> -->
        <fieldset class="starability-basic">
            <input type="radio" id="no-rate" class="input-no-rate" name="review[rating]"
value="1" checked
                aria-label="No rating." />
            <input type="radio" id="first-rate1" name="review[rating]" value="1" />
            <label for="first-rate1" title="Terrible">1 star</label>
```

```
            <input type="radio" id="first-rate2" name="review[rating]" value="2" />
            <label for="first-rate2" title="Not good">2 stars</label>
            <input type="radio" id="first-rate3" name="review[rating]" value="3" />
            <label for="first-rate3" title="Average">3 stars</label>
            <input type="radio" id="first-rate4" name="review[rating]" value="4" />
            <label for="first-rate4" title="Very good">4 stars</label>
            <input type="radio" id="first-rate5" name="review[rating]" value="5" />
            <label for="first-rate5" title="Amazing">5 stars</label>
        </fieldset>
        <div class="mb-3">
            <label class="form-label" for="body">Review Text</label>
            <textarea class="form-control" name="review[body]" id="body" cols="30"
rows="3" required></textarea>
            <div class="valid-feedback">
                Looks good!
            </div>
        </div>
        <button class="btn btn-success">Submit</button>
    </form>
    <% } %>
    <% for(let review of campground.reviews) { %>
    <div class="card mb-3 ">
        <div class="card-body">
            <h5 class="card-title"> <%= review.author.username%></h5>
            <p class="starability-result" data-rating="<%=review.rating%>">
                Rated: <%= review.rating %> stars
            </p>
            <!-- <h6 class="card-subtitle mb-2 text-muted">By <%=
review.author.username%></h6> -->
            <p class="card-text">Review: <%= review.body %></p>
```

```
                <%  if( currentUser && review.author.equals(currentUser._id))  {%>
                <form
    action="/campgrounds/<%=campground._id%>/reviews/<%=review._id%>?_method=DE
    LETE" method="POST">
                    <button class="btn btn-sm btn-danger">Delete</button>
                </form>
                <% } %>
              </div>
            </div>
            <% } %>
        </div>
    </div>

    <script>
        const mapToken = '<%-process.env.MAPBOX_TOKEN%>';
        const campground = <%- JSON.stringify(campground) %>
    </script>

    <script src="/javascripts/showPageMap.js"></script>
```

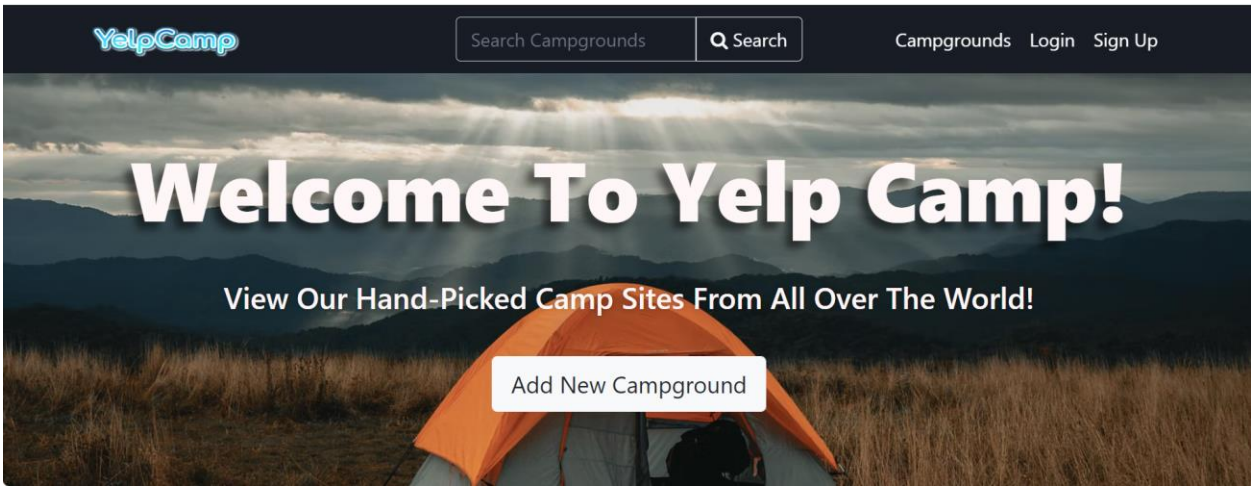
Index.ejs

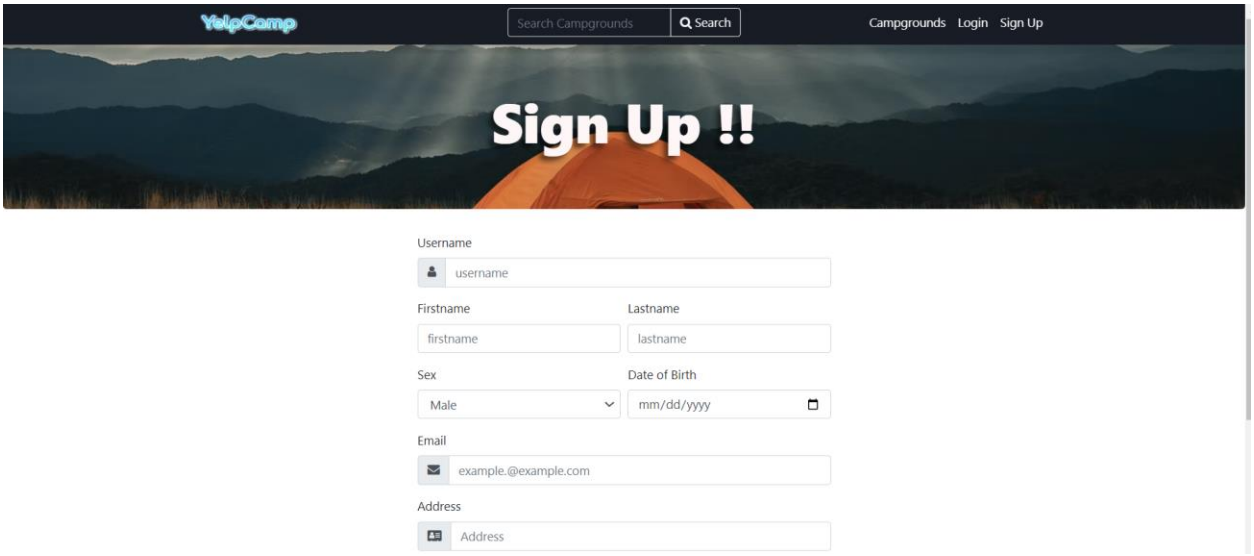
```
<div
id="clust
er-
map"></
div>
                <div class="container">
                    <h1>All Campgrounds</h1>
```

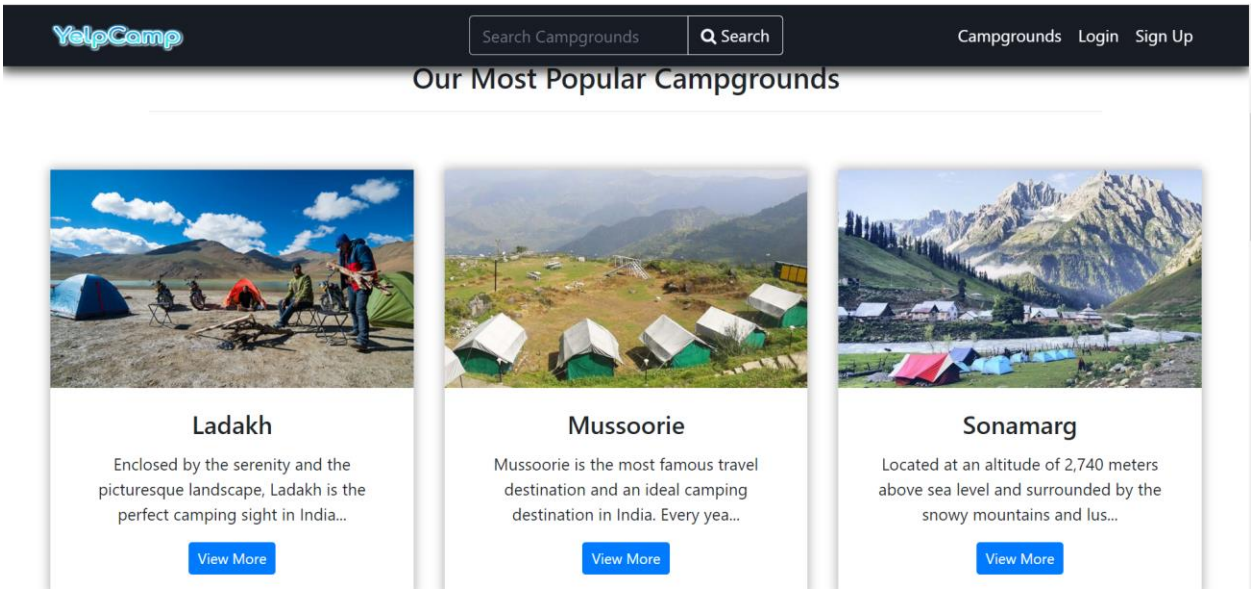
```
    <% for (let campground of campgrounds){%>
    <div class="card mb-3">
        <div class="row">
            <div class="col-md-4">
                <%if(campground.images.length) {%>
                <img class="img-fluid" alt=""
src="<%=campground.images[0].url%>">
                <% }else {%>
                <img class="img-fluid" alt=""

src="https://res.cloudinary.com/douqbebwk/image/upload/v1600103881/YelpCa
mp/lz8jjv2gyynjil7lswf4.png">
                <% } %>
            </div>
            <div class="col-md-8">
                <div class="card-body">
                    <h5 class="card-title"><%= campground.title %> </h5>
                    <p class="card-text"><%= campground.description %></p>
                    <p class="card-text">
                        <small class="text-muted"><%=
campground.location%></small>
                    </p>
                    <a class="btn btn-primary"
href="/campgrounds/<%=campground._id%>">View
<%=campground.title%></a>
                </div>
            </div>
        </div>
    </div>
    <% }%>
```

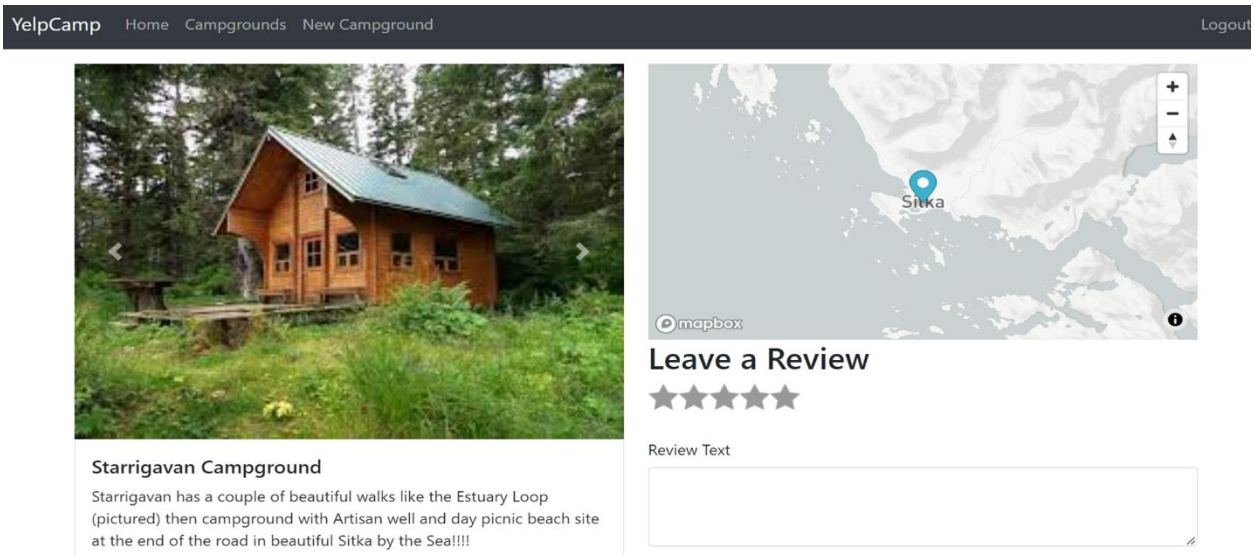
```
</div>
<script>
    const mapToken = '<%-process.env.MAPBOX_TOKEN%>';
    const campgrounds = { features: <%- JSON.stringify(campgrounds) %>}
</script>

<script src="/javascripts/clusterMap.js"></script>
```

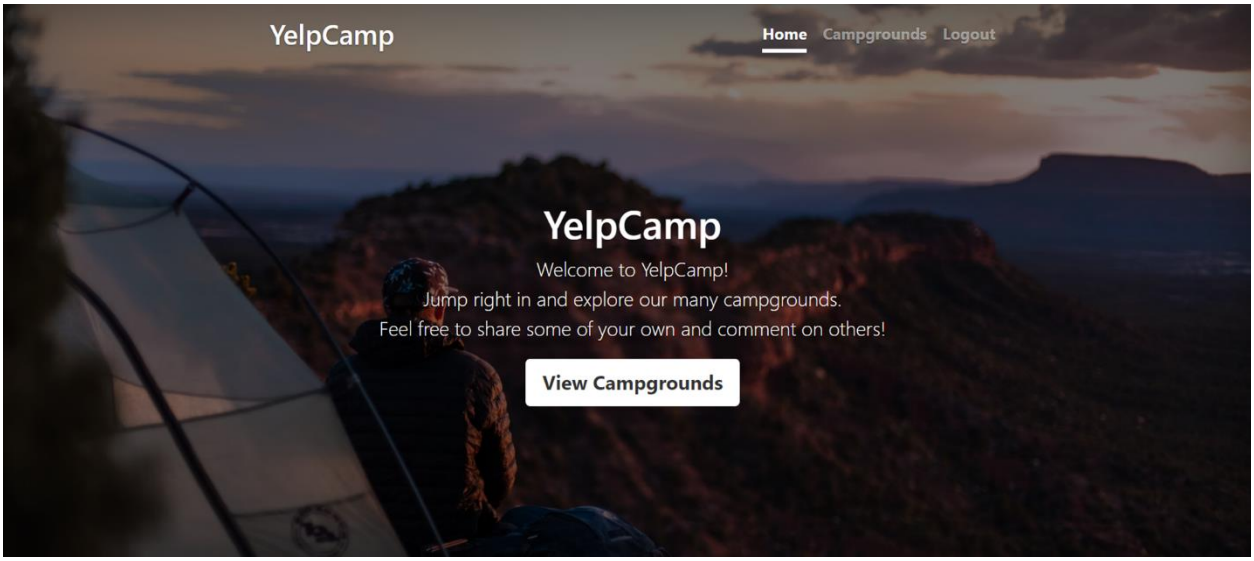
8. Screenshots

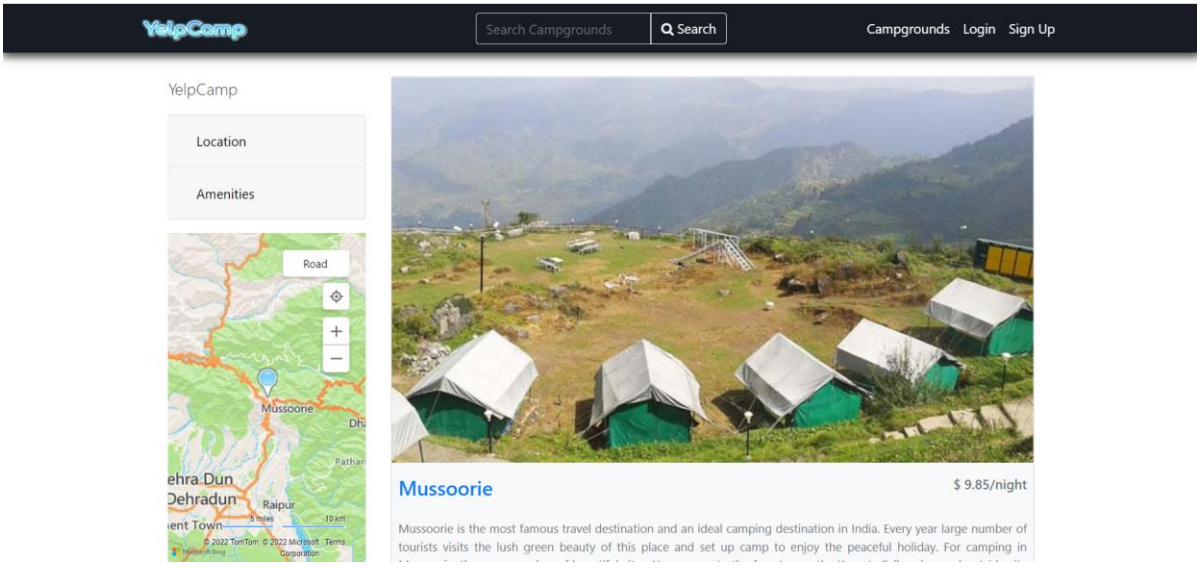





YelpCamp
Search Campgrounds
Search
Campgrounds
Login
Sign Up
Our Most Popular Campgrounds
Ladakh
Enclosed by the serenity and the picturesque landscape, Ladakh is the perfect camping sight in India...
View More
Mussoorie
Mussoorie is the most famous travel destination and an ideal camping destination in India. Every yea...
View More
Sonamarg
Located at an altitude of 2,740 meters above sea level and surrounded by the snowy mountains and lus...
View More


YelpCamp
Home
Campgrounds
New Campground
Logout
Sitka
mapbox
Leave a Review
Review Text
Starrigavan Campground
Starrigavan has a couple of beautiful walks like the Estuary Loop (pictured) then campground with Artisan well and day picnic beach site at the end of the road in beautiful Sitka by the Sea!!!!


YelpCamp
Home
Campgrounds
Logout
YelpCamp
Welcome to YelpCamp!
Jump right in and explore our many campgrounds.
Feel free to share some of your own and comment on others!
View Campgrounds


YelpCamp
Search Campgrounds
Search
Campgrounds
Login
Sign Up
YelpCamp
Location
Amenities
Road
Mussoorie
Dehradun
Raipur
Mussoorie
$ 9.85/night
Mussoorie is the most famous travel destination and an ideal camping destination in India. Every year large number of tourists visits the lush green beauty of this place and set up camp to enjoy the peaceful holiday. For camping in

9. FUTURE SCOPE

This project is build with extension in mind and can be extended to full fledged campgrounds booking application with the rating and review functionality remain intact. Altering users according to their requirement of the place and the price can be also be implement to enhance the functionality of the system. The module related to booking will help in monetizing this platform. Also, this can be extended to provide tour packages in some fixed price with all the accomodation, travel and other expenses included.

10. CONCLUSION

The YelpCamp Application is based on dynamic flow and rendering of data. An end user can register and use the application to store, explore and track the places based on the choice. The data is being saved to the database and location is saved using map coordinates.
It also has features related to any particular venue which further helps an end user to explore and find out more about the place. Hence, the application is responsive, dynamic, optimized and built using the best programming practices. Scaling the database and server, caching the application and effective use of memory further makes it fast.

11. REFERENCES

- https://www.w3schools.com/js/js_async.asp
- https://getbootstrap.com/docs/5.0/getting-started/introduction/
- https://stackoverflow.com/questions/tagged/reactjs
- https://1ustub.axshare.com/#g=1&p=splash_screen&id=cux4mo
- https://www.npmjs.com/package/mongoose
- https://www.free-css.com/template-categories/cafe-or-restaurant
- https://www.w3schools.com/js/js_promise.asp
- https://www.w3schools.com/js/js_callback.asp
- https://codepen.io/collection/KuDsH/