

Sri Lanka Institute of Information Technology



Final Exam

Web Solution for Event Planning System

Y3S2.09(WE)

ID NO: IT18003888

Application Framework [SE3040]

B.Sc. (Hons) in Computer Science and Software Engineering

CONTENT

- 1. Describe the scenario using your own words.**
- 2. Implementation of the main three functionalities (both frontend and backend).**
- 3. At least one implemented RESTful web service (implemented code).**
- 4. At least one Mongo query for a specific Mongo collection (implemented code).**
- 5. At least one-unit test using JEST to evaluate a core logic or a service (implemented code).**
- 6. A screenshot of the home page of the running application on localhost.**

1.0 Introduction

Describe the scenario

The Event Planner/admin supports event types. They are like Party, Conference and Birthdays and so on. The user can login to the system with user credentials. If user does not register for the system, he can register to the system by filling sign up page. If user logged to the system successfully, there is a pop up message will displayed and saying login successful. The user can view events in the beginning and the rest of the detailed. Apart from that in this scenario Mainly focus about admin duties. Admin can create event categories as required. Then admin can enter details furthermore. When creating a event, admin does not need to add again and again same event type, he can choose event type by dropdown. fields: Description about the event, location, capacity and all. Those added details will be displayed in a separate page. Then admin can edit those details and he can delete the event if necessary. When admin click on the edit button it will navigate to event page. Same process will be happened as edit when admin want to delete an event. Above described System mainly focus on Admin duties.

Furthermore, there can be more implementation in customer side.

For example, if it is a Party, the system will ask if music, party decorations or balloons should be present. All of these inputs will help to determine what kind of an event it should be. If all are in order, the system will calculate the cost based on the event type, venue, food type and other facilities and display the total cost to the user. If the user agrees with, he price, they can select confirm to book the event. The system will then add a record into the database with the user's selections. The specified date, time and venue will be booked and the users who would book after, would not be able to select this specific combination of date, time and venue.



2.0 Implementation of the main three functionalities.

- Customer can register and logging to the system.
- Admin can add, update, view, delete events.
- Customer can view added events.

Frontend

create-event.component.js

```
import React, { Component } from 'react';
import axios from 'axios';
import App from '../App.css';

//create event
export default class CreateEvent extends Component {

  //constructor
  constructor(props){
    super(props);

    this.onChangeUsername = this.onChangeUsername.bind(this);
    this.onSubmit = this.onSubmit.bind(this);

    this.state = {
      username: '',
    }
  }

  //method for change event name
  onChangeUsername(e){
    this.setState({
      username: e.target.value
    });
  }

  //onsubmit method
  onSubmit(e){
```

```
e.preventDefault();

const user = {
  username: this.state.username,
}

console.log(user);

axios.post('http://localhost:5000/users/add',user)
  .then(res => console.log(res.data));

this.setState({
  username: ""
})

//redirect to the create event page
window.location = '/create';
}

render(){
  return(
    <div>
      <div class="row">
        <div className="column">
          <br/>
        </div></div>

        //form create new event
        <h3>Create new Event</h3>
        <form onSubmit={this.onSubmit}>
          <div className="form-group">
            <label>EventName: </label>
            <input type="text"
              required
              className="form-control"
              value={this.state.username}
              onChange={this.onChangeUsername}
            />
          </div>

          //submit button
          <div className="form-group">
            <input type="submit" value="create Event" className="btn btn-primary" />
          </div>
        </div>
```

```
        </form>
      </div>
    )
  }
}
```

create-eventexercise.component.js

```
import React, { Component } from 'react';
import axios from 'axios';
import DatePicker from 'react-datepicker';
import "react-datepicker/dist/react-datepicker.css";

//create event exercise
export default class CreateExercise extends Component {
  constructor(props) {
    super(props);

    this.onChangeUsername = this.onChangeUsername.bind(this);
    this.onChangeDescription = this.onChangeDescription.bind(this);
    this.onChangeDuration = this.onChangeDuration.bind(this);
    this.onChangeLocation = this.onChangeLocation.bind(this);
    this.onChangeDate = this.onChangeDate.bind(this);
    this.onSubmit = this.onSubmit.bind(this);

    this.state = {
      username: "",
      description: "",
      duration: 0,
      location: "",
      date: new Date(),
      users: []
    }
  }

  //componentDidMount
  componentDidMount() {
```

```
axios.get('http://localhost:5000/users/')
  .then(response => {
    if (response.data.length > 0) {
      this.setState({
        users: response.data.map(user => user.username),
        username: response.data[0].username
      })
    }
  })
  .catch((error) => {
    console.log(error);
  })
}

//method for form attributes
onChangeUsername(e) {
  this.setState({
    username: e.target.value
  })
}

onChangeDescription(e) {
  this.setState({
    description: e.target.value
  })
}

onChangeDuration(e) {
  this.setState({
    duration: e.target.value
  })
}

onChangeLocation(e) {
  this.setState({
    location: e.target.value
  })
}

onChangeDate(date) {
  this.setState({
    date: date
  })
}
```

```
//onSubmit
onSubmit(e) {
  e.preventDefault();

  const exercise = {
    username: this.state.username,
    description: this.state.description,
    duration: this.state.duration,
    location: this.state.location,
    date: this.state.date
  }

  console.log(exercise);

  axios.post('http://localhost:5000/exercises/add', exercise)
    .then(res => console.log(res.data));

  //redirect to the home page
  window.location = '/';
}

render() {
  return (
    <div>
      <h3>Create New Event Log</h3>
      //form
      <form onSubmit={this.onSubmit}>
        <div className="form-group">
          <label>EventName: </label>
          <select ref="userInput"
            required
            className="form-control"
            value={this.state.username}
            onChange={this.onChangeUsername}>
            {
              this.state.users.map(function(user) {
                return <option
                  key={user}
                  value={user}>{user}
                </option>;
              })
            }
          </select>
        </div>
        <div className="form-group">
          <label>Description: </label>
          <input type="text"
            required
            className="form-control"
            value={this.state.description}
            onChange={this.onChangeDescription}
          />
        </div>
      </form>
    </div>
  );
}
```



```
</div>
<div className="form-group">
  <label>Capacity: </label>
  <input
    type="text"
    className="form-control"
    value={this.state.duration}
    onChange={this.onChangeDuration}
  />
</div>

<div className="form-group">
  <label>Location </label>
  <input
    type="text"
    className="form-control"
    value={this.state.locationn}
    onChange={this.onChangeLocation}
  />
</div>

<div className="form-group">
  <label>Date: </label>
  <div>
    <DatePicker
      selected={this.state.date}
      onChange={this.onChangeDate}
    />
  </div>
</div>

<div className="form-group">
  <input type="submit" value="Create Exercise Log" className="btn btn-primary" />
</div>
</form>
</div>
)
}
```

edit-eventexercise.component.js

```
import React, { Component } from 'react';
import axios from 'axios';
import DatePicker from 'react-datepicker';
import "react-datepicker/dist/react-datepicker.css";

export default class EditExercise extends Component {
```

```
constructor(props) {
  super(props);

  this.onChangeUsername = this.onChangeUsername.bind(this);
  this.onChangeDescription = this.onChangeDescription.bind(this);
  this.onChangeDuration = this.onChangeDuration.bind(this);
  this.onChangeLocation = this.onChangeLocation.bind(this);
  this.onChangeDate = this.onChangeDate.bind(this);
  this.onSubmit = this.onSubmit.bind(this);

  this.state = {
    username: "",
    description: "",
    duration: 0,
    location: "",
    date: new Date(),
    users: []
  }
}

//componentDidMount
componentDidMount() {
  axios.get('http://localhost:5000/exercises/' + this.props.match.params.id)
    .then(response => {
      this.setState({
        username: response.data.username,
        description: response.data.description,
        duration: response.data.duration,
        location: response.data.location,
        date: new Date(response.data.date)
      })
    })
    .catch(function (error) {
      console.log(error);
    })
}

//get the data from backend
axios.get('http://localhost:5000/users/')
  .then(response => {
    if (response.data.length > 0) {
      this.setState({
        users: response.data.map(user => user.username),
      })
    }
  })
  .catch((error) => {
    console.log(error);
  })
}

onChangeUsername(e) {
```

```
this.setState({
  username: e.target.value
})
}

onChangeDescription(e) {
  this.setState({
    description: e.target.value
  })
}

onChangeDuration(e) {
  this.setState({
    duration: e.target.value
  })
}

onChangeLocation(e) {
  this.setState({
    location: e.target.value
  })
}

onChangeDate(date) {
  this.setState({
    date: date
  })
}

//onSubmit
onSubmit(e) {
  e.preventDefault();

  const exercise = {
    username: this.state.username,
    description: this.state.description,
    duration: this.state.duration,
    location: this.state.location,
    date: this.state.date
  }

  console.log(exercise);

  axios.post('http://localhost:5000/exercises/update/' + this.props.match.params.id, exercise)
    .then(res => console.log(res.data));

  window.location = '/';
}

render() {
  return (
    <div>
```

```
<h3>Edit Event Log</h3>
<form onSubmit={this.onSubmit}>
  <div className="form-group">
    <label>EventName: </label>
    <select ref="userInput"
      required
      className="form-control"
      value={this.state.username}
      onChange={this.onChangeUsername}>
      {
        this.state.users.map(function(user) {
          return <option
            key={user}
            value={user}>{user}
          </option>;
        })
      }
    </select>
  </div>
  <div className="form-group">
    <label>Description: </label>
    <input type="text"
      required
      className="form-control"
      value={this.state.description}
      onChange={this.onChangeDescription}
    />
  </div>
  <div className="form-group">
    <label>Capacity: </label>
    <input
      type="text"
      className="form-control"
      value={this.state.duration}
      onChange={this.onChangeDuration}
    />
  </div>

  <div className="form-group">
    <label>Location: </label>
    <input
      type="text"
      className="form-control"
      value={this.state.location}
      onChange={this.onChangeLocation}
    />
  </div>

  <div className="form-group">
    <label>Date: </label>
    <div>
      <DatePicker
```

```
        selected={this.state.date}
        onChange={this.onChangeDate}
      />
    </div>
  </div>

  <div className="form-group">
    <input type="submit" value="Edit Event Log" className="btn btn-primary" />
  </div>
</form>
</div>
)
}
```

eventexercises-list.component.js

```
import React, { Component } from 'react';
import { Link } from 'react-router-dom';
import axios from 'axios';

const Exercise = props => (

  //list - to view data
  <tr>
    <td>{props.exercise.username}</td>
    <td>{props.exercise.description}</td>
    <td>{props.exercise.duration}</td>
    <td>{props.exercise.location}</td>
    <td>{props.exercise.date.substring(0,10)}</td>
    <td>
      <Link to={"/edit/"+props.exercise._id}>edit</Link> | <a href="#" onClick={() => {
        props.deleteExercise(props.exercise._id) }}>delete</a>
    </td>
  </tr>
)

export default class ExercisesList extends Component {
  constructor(props) {
    super(props);

    this.deleteExercise = this.deleteExercise.bind(this)

    this.state = {exercises: []};
  }

  //componentdidMount
  componentDidMount() {
```

```
axios.get('http://localhost:5000/exercises/')
  .then(response => {
    this.setState({ exercises: response.data })
  })
  .catch((error) => {
    console.log(error);
  })
}

//delete
deleteExercise(id) {
  axios.delete('http://localhost:5000/exercises/'+id)
    .then(response => { console.log(response.data)});

  this.setState({
    exercises: this.state.exercises.filter(el => el._id !== id)
  })
}

//list
exerciseList() {
  return this.state.exercises.map(currentexercise => {
    return <Exercise exercise={currentexercise} deleteExercise={this.deleteExercise}
    key={currentexercise._id}/>;
  })
}

render() {
  return (
    <div>
      <h3>Events</h3>
      <table className="table">
        <thead className="thead-light">
          <tr>
            <th>EventName</th>
            <th>Description</th>
            <th>Capacity</th>
            <th>Location</th>
            <th>Date</th>
            <th>Actions</th>
          </tr>
        </thead>
        <tbody>
          { this.exerciseList() }
        </tbody>
      </table>
    </div>
  )
}
```

home.js

```
import React, { Component } from 'react';
//import { Link } from 'react-router-dom';
import axios from 'axios';
import '../App.css';

const Exercise = props => (
  <tr>
    <td>{props.exercise.username}</td>
    <td>{props.exercise.description}</td>
    <td>{props.exercise.duration}</td>
    <td>{props.exercise.location}</td>
    <td>{props.exercise.date.substring(0,10)}</td>
    { /* <td>
      <Link to={"/edit/"+props.exercise._id}>edit</Link> | <a href="#" onClick={() => {
props.deleteExercise(props.exercise._id) }}>delete</a>
      </td> */ }
    </tr>
  )

export default class ExercisesList extends Component {
  constructor(props) {
    super(props);

    //this.deleteExercise = this.deleteExercise.bind(this)

    this.state = { exercises: [] };
  }

  //componentDidMount
  componentDidMount() {
    axios.get('http://localhost:5000/exercises/')
      .then(response => {
        this.setState({ exercises: response.data })
      })
      .catch((error) => {
        console.log(error);
      })
  }

  // deleteExercise(id) {
  //   axios.delete('http://localhost:5000/exercises/'+id)
  //   .then(response => { console.log(response.data)});

  //   this.setState({
  //     exercises: this.state.exercises.filter(el => el._id !== id)
  //   })
  // }
```

```
exerciseList() {  
  return this.state.exercises.map(currentexercise => {  
    return <Exercise exercise={currentexercise} deleteExercise={this.deleteExercise}  
key={currentexercise._id}/>;  
  })  
}  
  
render() {  
  return (  
    <div>  
      <h3>ReSerVe yOur TiMe</h3>  
  
      <div>  
          
  
      </div>  
  
      <div>  
          
  
      </div>  
  
      <br/><br/>  
      <table className="table">  
        <thead className="thead-light">  
          <tr>  
            <th>EventName</th>  
            <th>Description</th>  
            <th>Capacity</th>  
            <th>Location</th>  
            <th>Date</th>  
  
          </tr>  
        </thead>  
        <tbody>  
          { this.exerciseList() }  
        </tbody>  
      </table>  
    </div>  
  )  
}
```


LoginTab.js

```
import React from 'react';
import { Paper, withStyles } from '@material-ui/core';
import axios from 'axios';
import Swal from 'sweetalert2/dist/sweetalert2.js';

import 'sweetalert2/src/sweetalert2.scss';

const styles = theme => ({
  margin: {
    margin: theme.spacing.unit * 2,
  },
  padding: {
    padding: theme.spacing.unit
  }
});

const customStyles = {
  content: {
    top: '38%',
    left: '50%',
    right: 'auto',
    bottom: 'auto',
    width: 800,
    marginRight: '-50%',
    transform: 'translate(-50%, -50%)',
    backgroundColor: '#deffd4'
  }
};

class LoginTab extends React.Component {

  constructor(props){
    super(props);

    this.onChangeemail = this.onChangeemail.bind(this);
    this.onChangepassword = this.onChangepassword.bind(this);
    this.onsubmit = this.onsubmit.bind(this);

    this.state = {
      email: "",
      password: ""
    }
  }

  onChangeemail(e){
    this.setState({
      email: e.target.value
    });
  }
}
```

```
onChangepassword(e){
  this.setState({
    password: e.target.value
  });
}

onsubmit(e){
  e.preventDefault();

  const user = {
    email: this.state.email,
    password: this.state.password
  }

  console.log(user);

  axios.post('http://localhost:5000/users/login', user)
    .then(res => console.log(res.data));
  Swal.fire({
    position: 'center',
    icon: 'success',
    title: 'Your work has been saved',
    showConfirmButton: false,
    timer: 10500
  })

  this.setState({
    username: ''
  })

  window.location = "/home";
}

render() {
  const { classes } = this.props;

  return (
    <div style={{height: 500}}>
      <div className="container">
        <Paper className={classes.padding} style={{paddingTop: 20, marginTop: 40}}>
          <h3 style={{textAlign: "center"}}>Login</h3><br/>
          <div className="container">
            <div className="form-group">
              <label for="inputAddress">Username</label>
              <input type="text" className="form-control" id="inputAddress"
onChange={this.onChangeemail} value={this.state.email}/>
            </div>
            <div className="form-group">
              <label for="inputAddress2">Password</label>
              <input type="password" className="form-control" id="inputAddress2"
onChange={this.onChangepassword} value={this.state.password} />
            </div>
          </div>
        </Paper>
      </div>
    </div>
  );
}
```

```
    <div className="form-group">
      <div className="form-check">
        <input className="form-check-input" type="checkbox" id="gridCheck"/>
        <label className="form-check-label" for="gridCheck">
          Remember Me
        </label>
      </div>
    </div>
    <button type="submit" className="btn btn-success btn-lg" style={{marginLeft: 490}}
onClick={this.onSubmit}>Submit</button>
  </div>
</Paper>
</div>
</div>
);
}
}

export default withStyles(styles)(LoginTab);
```

Signup.js

```
import React from 'react';
import { Paper, withStyles } from '@material-ui/core';
import axios from 'axios';

const styles = theme => ({
  margin: {
    margin: theme.spacing.unit * 2,
  },
  padding: {
    padding: theme.spacing.unit
  }
});

class Signup extends React.Component {

  //constructor
  constructor(props){
    super(props);

    this.onChangefirstName = this.onChangefirstName.bind(this);
    this.onChangelastName = this.onChangelastName.bind(this);
    this.onChangeemail = this.onChangeemail.bind(this);
    this.onChangepassword = this.onChangepassword.bind(this);
    this.onSubmit = this.onSubmit.bind(this);

    this.state = {
      firstName: "",
      lastName: "",
```

```
    email: "",
    password: ""
  }
}

onChangefirstName(e){
  this.setState({
    firstName: e.target.value
  });
}

onChangelastName(e){
  this.setState({
    lastName: e.target.value
  });
}

onchangeemail(e){
  this.setState({
    email: e.target.value
  });
}

onChangepassword(e){
  this.setState({
    password: e.target.value
  });
}

//onSubmit
onsubmit(e){
  console.log(this.state.firstName+" this is name")
  e.preventDefault();

  const user = {
    firstName: this.state.firstName,
    lastName: this.state.lastName,
    email: this.state.email,
    password: this.state.password
  }

  console.log(user);

  //get data
  axios.post('http://localhost:5000/users/signup', user)
    .then(res => console.log(res.data));

  this.setState({
    username: ""
  })
}
```

```
//redirect login
window.location = "/login";
}
render() {
  const { classes } = this.props;

  return (
    <div style={{height: 500}}>
      <div className="container">
        <Paper className={classes.padding} style={{paddingTop: 10, marginTop: 40}}>
          <h3 style={{textAlign: "center"}}>Sign Up</h3><br/>
          <div className="container">
            <div class="form-row">
              <div className="form-group col-md-6">
                <label for="inputEmail4">First Name</label>
                <input type="email" className="form-control" placeholder="Brad" id="inputEmail4"
onChange={this.onChangefirstName} value={this.state.firstName}/>
              </div>
              <div className="form-group col-md-6">
                <label for="inputPassword4">Last Name</label>
                <input type="text" className="form-control" placeholder="Gibson" id="inputLastname"
onChange={this.onChangelastName} value={this.state.lastName}/>
              </div>
            </div>
            <div className="form-group">
              <label for="inputAddress">Email Address</label>
              <input type="text" className="form-control" id="inputAddress"
placeholder="abc@gmail.com" onChange={this.onChangeemail} value={this.state.email}/>
            </div>
            <div className="form-group">
              <label for="inputAddress2">Password</label>
              <input type="password" className="form-control" id="inputAddress2"
onChange={this.onChangepassword} value={this.state.password}/>
            </div>

            <div className="form-group">
              <div className="form-check">
                <input className="form-check-input" type="checkbox" id="gridCheck"/>
                <label className="form-check-label" for="gridCheck">
                  Check me out
                </label>
              </div>
            </div>
            <button type="submit" className="btn btn-primary btn-lg" style={{marginLeft: 480}}
onClick={this.onSubmit}>Register</button>
          </div>
        </Paper>
      </div>
    </div>
  );
}
```

```
}  
  
export default withStyles(styles)(Signup);
```

navbar.component.js

```
//const { Component } = require("react");  
  
import React, { Component } from 'react';  
import {Link} from 'react-router-dom';  
  
export default class Navbar extends Component{  
  
  //nav links  
  render(){  
    return(  
      <nav className="navbar navbar-dark bg-dark navbar-expand-lg">  
        <div>  
          <h3><Link to="/home" className="Navbar-brand">Event Planner</Link></h3>  
        </div>  
        <div className="collapse navbar-collapse">  
          <br/>  
          <ul className="navbar-nav mr-auto">  
            <li className="navbar-item">  
              <Link to="/" className="nav-link">Events</Link>  
            </li><br/><br/>  
            <li className="navbar-item">  
              <Link to="/create" className="nav-link">Create Event Logs</Link>  
            </li><br/><br/>  
            <li className="navbar-item">  
              <Link to="/user" className="nav-link">Create Event</Link>  
            </li>  
          </ul>  
  
          <li className="navbar-item">  
            <Link to="/signup" className="nav-link">Sign Up</Link>  
          </li><br/><br/>  
  
          <li className="navbar-item">  
            <Link to="/login" className="nav-link">Sign In</Link>  
          </li><br/><br/>  
        </div>  
      </nav>  
    )  
  }  
}
```

App.js

```
import React from 'react';
import {BrowserRouter as Router, Route} from 'react-router-dom';
import "bootstrap/dist/css/bootstrap.min.css"

import Navbar from "../components/navbar.component";
import ExercisesList from "../components/eventexercises-list.component";
import EditExercise from "../components/edit-eventexercise.component";
import CreateExercise from "../components/create-eventexercise.component";
import CreateUser from "../components/create-event.component";
import Home from "../components/home";
import Signup from "../components/Signup";
import LoginTab from "../components/LoginTab";

function App() {
  return (
    <Router>
      <div className="container">
        <Navbar/>
        <br/>
        <Route path="/home" exact component={Home} />
        <Route path="/" exact component={ExercisesList} />
        <Route path="/edit/:id" exact component={EditExercise} />
        <Route path="/create" exact component={CreateExercise} />
        <Route path="/user" exact component={CreateUser} />
        <Route path="/signup" exact component={Signup} />
        <Route path="/login" exact component={LoginTab} />
      </div>
    </Router>
  );
}

export default App;
```

Backend**Server.js**

```
const express = require('express');
const cors = require('cors');
const mongoose = require('mongoose');

//environment variable
require('dotenv').config();

//express server
const app = express();
const port = process.env.PORT || 5000;

//middleware

app.use(cors());
app.use(express.json());

//mongo cloud connection
const uri = process.env.ATLAS_URI;
mongoose.connect(uri,{ useNewUrlParser: true, useCreateIndex: true, useUnifiedTopology: true }
);

const connection = mongoose.connection;
connection.once('open', () => {
  console.log("MongoDB database connection established successfully");
})

const exercisesRouter = require('./routes/event');
const usersRouter = require('./routes/users');
//const usersRoute = require('./routes/users.router');
//const userLoginRoute = require('./routes/usersLogin.router');

app.use('/exercises', exercisesRouter);
app.use('/users', usersRouter);
// app.use('/Users',usersRoute);
// app.use('/loginUser', userLoginRoute);

//start the server
app.listen(port, () => {
  console.log(`Server is running on port: ${port}`);
});
```


3.0 At least one implemented RESTful web service

user.js

//signup

```
router.post('/signup', (req, res) => {  
  const sign = new signup({  
    firstName: req.body.firstName,  
    lastName: req.body.lastName,  
    email: req.body.email,  
    password: req.body.password,  
  })  
})
```

```
sign.save().then(sign => {  
  console.log("signup successfully")  
  alert("SUCCESSFUL!");  
  try{  
    res.status(200).send({  
      message: 'sign up succesfully',  
      data: sign  
    })  
  }catch(err){  
    res.status(502).send({  
      message: 'Error!',  
      error: err  
    })  
  }  
})  
})
```

//login

```
router.post('/login', (req, res) => {  
  signup.findOne({  
    email: req.body.email,  
    password: req.body.password,  
  }).then(admin => {  
    if(admin) {  
      res.send({  
        message: 'Successfully logged in',  
        data: admin,  
        messageCode: "1000"  
      })  
    }else{  
      console.log("not found")  
      res.send({  
        message: 'Invalid credetials',  
        data: admin,  
        messageCode: "1001"  
      })  
    }  
  })  
})
```

event.js

```
const router = require('express').Router();
let Exercise = require('../models/event.model');

router.route('/').get((req, res) => {
  Exercise.find()
    .then(exercises => res.json(exercises))
    .catch(err => res.status(400).json('Error: ' + err));
});

//add
router.route('/add').post((req, res) => {
  const username = req.body.username;
  const description = req.body.description;
  const duration = Number(req.body.duration);
  const location = req.body.location;
  const date = Date.parse(req.body.date);

  const newExercise = new Exercise({
    username,
    description,
    duration,
    location,
    date,
  });

  newExercise.save()
    .then(() => res.json('Event added!'))
    .catch(err => res.status(400).json('Error: ' + err));

});

//get data by id
router.route('/:id').get((req, res) => {
  Exercise.findById(req.params.id)
    .then(exercise => res.json(exercise))
    .catch(err => res.status(400).json('Error: ' + err));
});

//delete data
router.route('/:id').delete((req, res) => {
  Exercise.findByIdAndDelete(req.params.id)
    .then(() => res.json('Event Deleted '))
    .catch(err => res.status(400).json('Error: ' + err));
});
```

```
//  
router.route('/update/:id').post((req, res) => {  
  Exercise.findById(req.params.id)  
  .then(exercise => {  
    exercise.username = req.body.username;  
    exercise.description = req.body.description;  
    exercise.duration = Number(req.body.duration);  
    exercise.location = req.body.location;  
    exercise.date = Date.parse(req.body.date);  
  
    exercise.save()  
    .then(() => res.json('Event Updated'))  
    .catch(err => res.status(400).json('Error: ' + err));  
  })  
  .catch(err => res.status(400).json('Error: ' + err));  
});  
  
module.exports = router;
```

4.0 At least one Mongo query for a specific Mongo collection

event.model.js

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

// create schema event
const Schema = mongoose.Schema;

const exerciseSchema = new Schema({
  username: {type: String, required: true},
  description: { type: String, required: true},
  duration: { type: Number, required: true},
  location: { type: String, required: true},
  date: { type: Date, required: true},
},{
  timestamps: true,
});

const Exercise = mongoose.model('Exercise',exerciseSchema);

module.exports = Exercise;
```

signup.js

```
const mongoose = require('mongoose')

//create schema signup
const schema = mongoose.Schema;

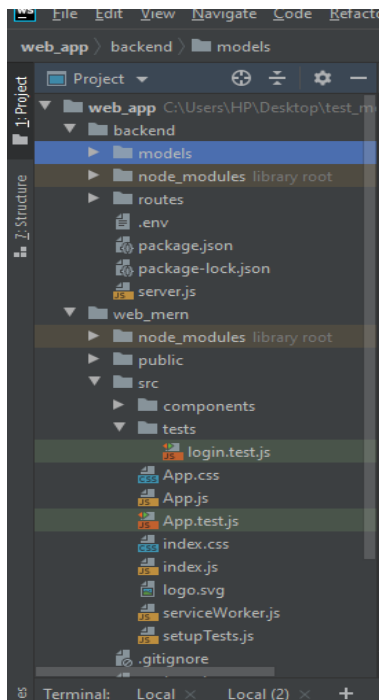
const signupSchema = new schema({
  firstName: {
    type: String,
    required: true,
    trim: true,
    minlength: 3
  },
  lastName: {
    type: String,
    required: true,
    trim: true,
    minlength: 3
  },
  email: {
    type: String,
    required: true,
    unique: true,
    trim: true,
    minlength: 3
  },
},
```

```
password: {  
  type: String,  
  required: true,  
  unique: true,  
  trim: true,  
  minlength: 3  
}  
, {  
  timestamps: true,  
});  
  
const signup = mongoose.model('signup', signupSchema);  
  
module.exports = signup;
```

.env

```
ATLAS_URI=mongodb+srv://dbuser:dbuser@cluster0-  
ylbgx.gcp.mongodb.net/<dbname>?retryWrites=true&w=majority
```

5.0 At least one unit test using JEST to evaluate a core logic or a service



login.test.js

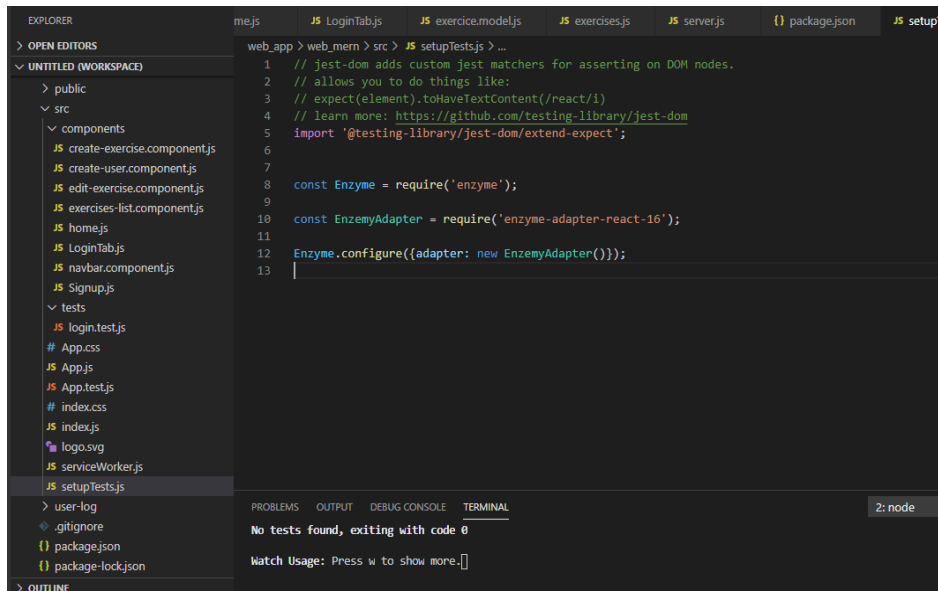
```
import React from 'react';
import {shallow} from 'enzyme';
import Login from '../components/LoginTab';

//testing withjest
describe('LoginTab Component', () => {
  it('should render without throwing an error', () => {
    expect(shallow(<Login />).find('form.LoginTab').exists()).toBe(true)
  })
})

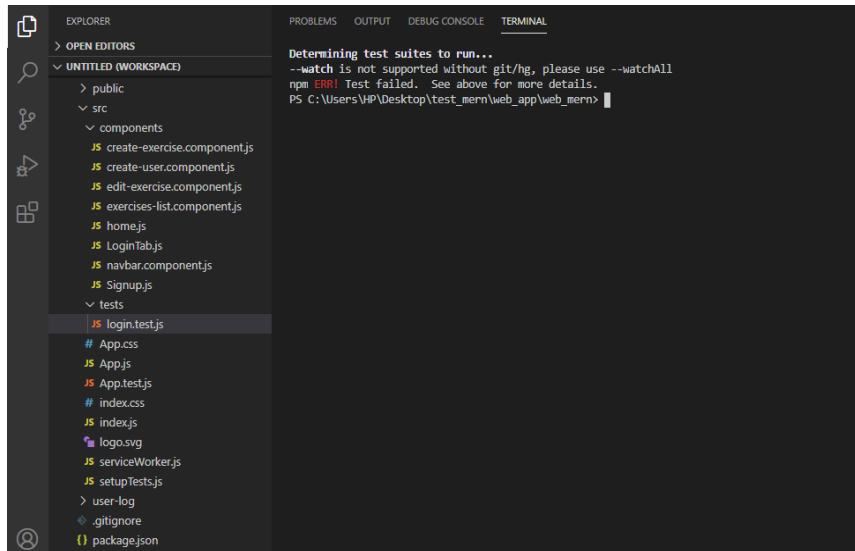
it('renders a email input', () => {
  expect(shallow(<Login />).find('#email').length).toEqual(1)
})
it('renders a password input', () => {
  expect(shallow(<Login />).find('#password').length).toEqual(1)
})
})
```

```
describe('Email input', () => {
  it('should respond to change event and change the state of the Login Component', () => {
    const wrapper = shallow(<Login />);
    wrapper.find('#email').simulate('change', {target: {name: 'email', value:
    'blah@gmail.com'}});
    expect(wrapper.state('email')).toEqual('blah@gmail.com');
  })
})
describe('Password input', () => {
  it('should respond to change event and change the state of the Login Component', () => {
    const wrapper = shallow(<Login />);
    wrapper.find('#password').simulate('change', {target: {name: 'password', value:
    'cats'}});
    expect(wrapper.state('password')).toEqual('cats');
  })
})
```

setupTests.js



Result:

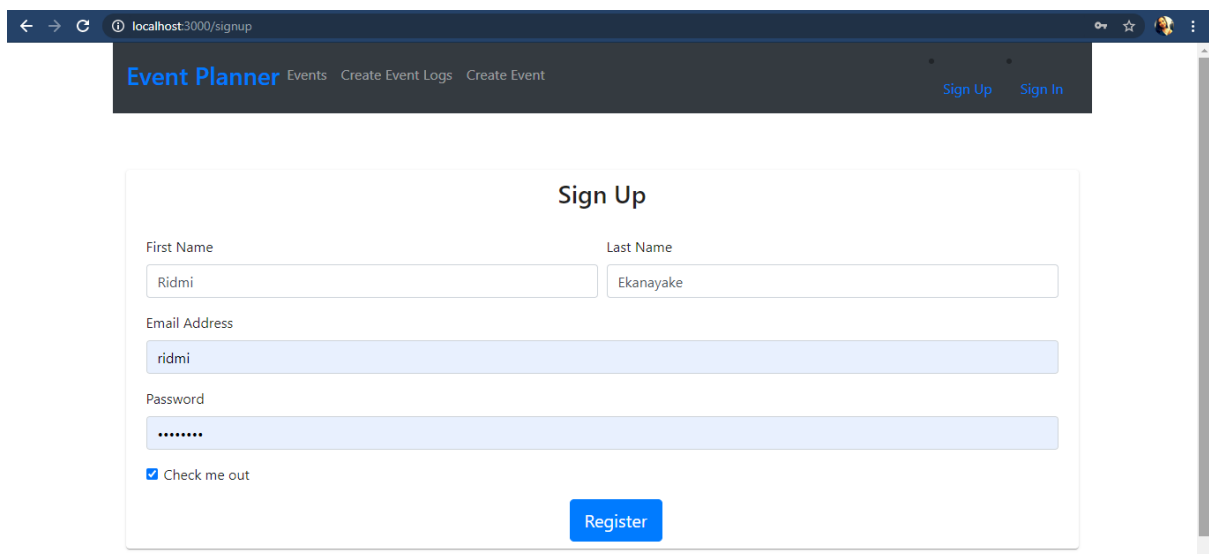


```
EXPLORER
> OPEN EDITORS
v UNTITLED (WORKSPACE)
  > public
  v src
    v components
      JS create-exercise.component.js
      JS create-user.component.js
      JS edit-exercise.component.js
      JS exercises-list.component.js
      JS home.js
      JS LoginTab.js
      JS navbar.component.js
      JS Signup.js
    v tests
      JS login.test.js
  App.css
  JS App.js
  JS App.test.js
  # index.css
  JS index.js
  logo.svg
  JS serviceWorker.js
  JS setupTests.js
  > user-log
  .gitignore
  package.json

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Determining test suites to run...
--watch is not supported without git/hg, please use --watchAll
npm ERR! Test failed.  See above for more details.
PS C:\Users\HP\Desktop\test_mern\web_app\web_mern>
```


6.0 A screenshot of the home page of the running application on localhost.

Sign up

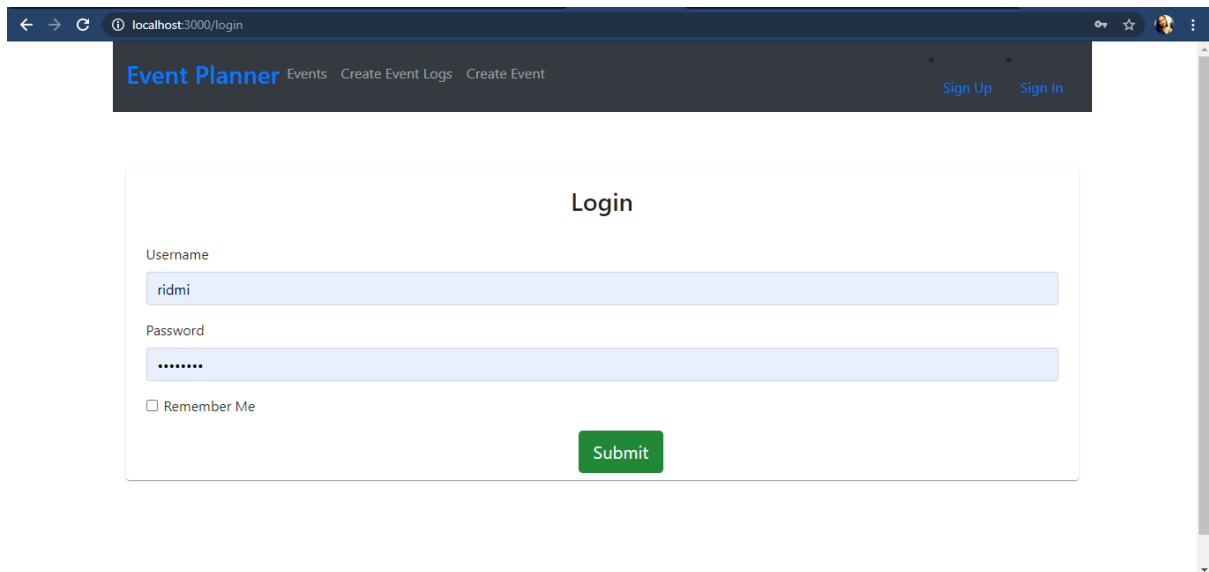


The screenshot shows a web browser window with the address bar displaying 'localhost:3000/signup'. The page has a dark blue header with the text 'Event Planner' and navigation links 'Events', 'Create Event Logs', and 'Create Event'. On the right side of the header, there are links for 'Sign Up' and 'Sign In'. The main content area is a white box titled 'Sign Up' containing the following form fields:

- First Name:** A text input field with the value 'Ridmi'.
- Last Name:** A text input field with the value 'Ekanayake'.
- Email Address:** A text input field with the value 'ridmi'.
- Password:** A password input field with masked characters '*****'.
- Check me out:** A checkbox that is checked.
- Register:** A blue button to submit the form.

Figure - Sign up page for customer

Customer can register to the system through the sign up interface.

Login

The screenshot shows a web browser window with the address bar displaying 'localhost:3000/login'. The page has a dark header with the text 'Event Planner' and navigation links 'Events', 'Create Event Logs', and 'Create Event'. On the right side of the header are links for 'Sign Up' and 'Sign In'. The main content area is a white box titled 'Login' containing a form. The form has two input fields: 'Username' with the value 'ridmi' and 'Password' with masked characters '.....'. Below the password field is a checkbox labeled 'Remember Me'. A green 'Submit' button is located at the bottom right of the form.

Figure – Login page

Customer who register to the system can login to the system use the credentials. Admin can login to the system with his own credentials , he doesn't want to register to the system.

Alert – login success

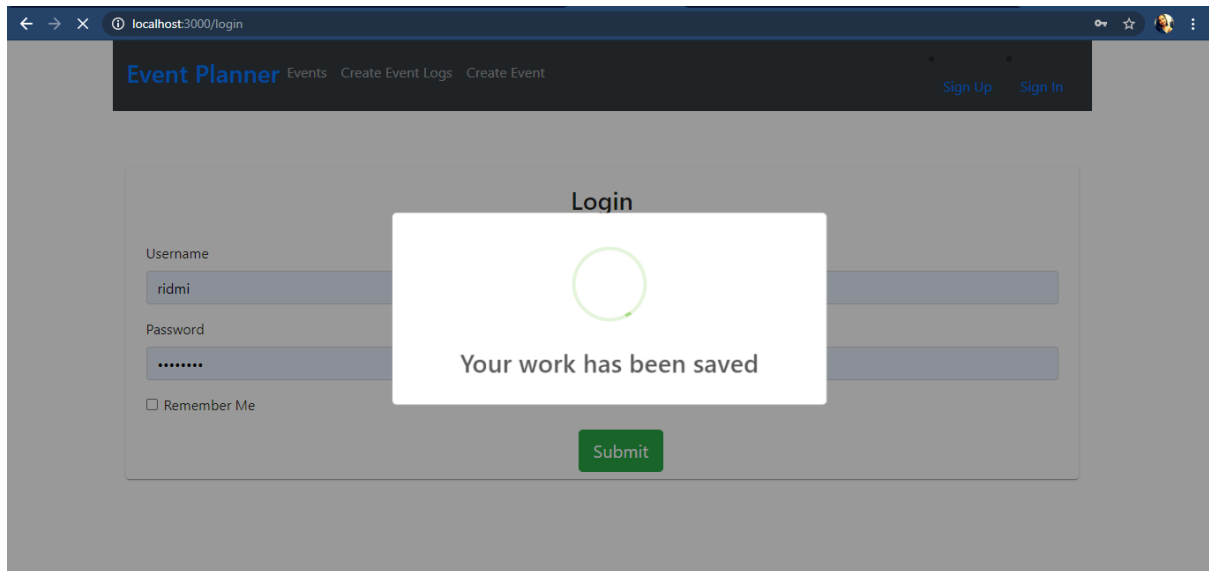


Figure – alert message

This alert message for successfully login to the system

Home

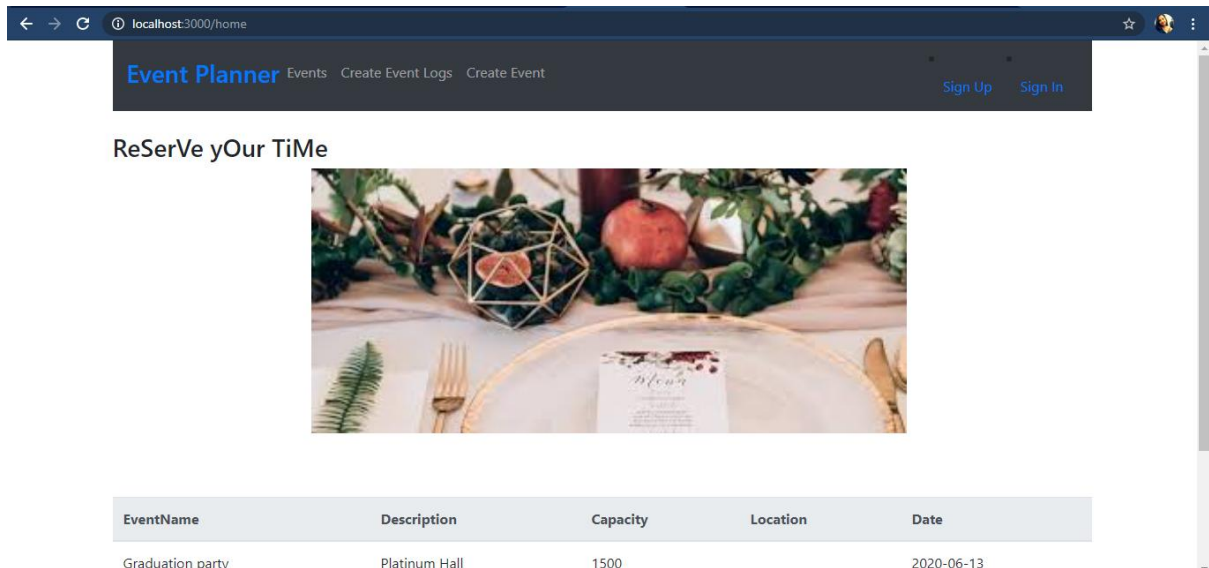


Figure – Home Page

Registered customer can view added events to the websites and get the information about the Events.

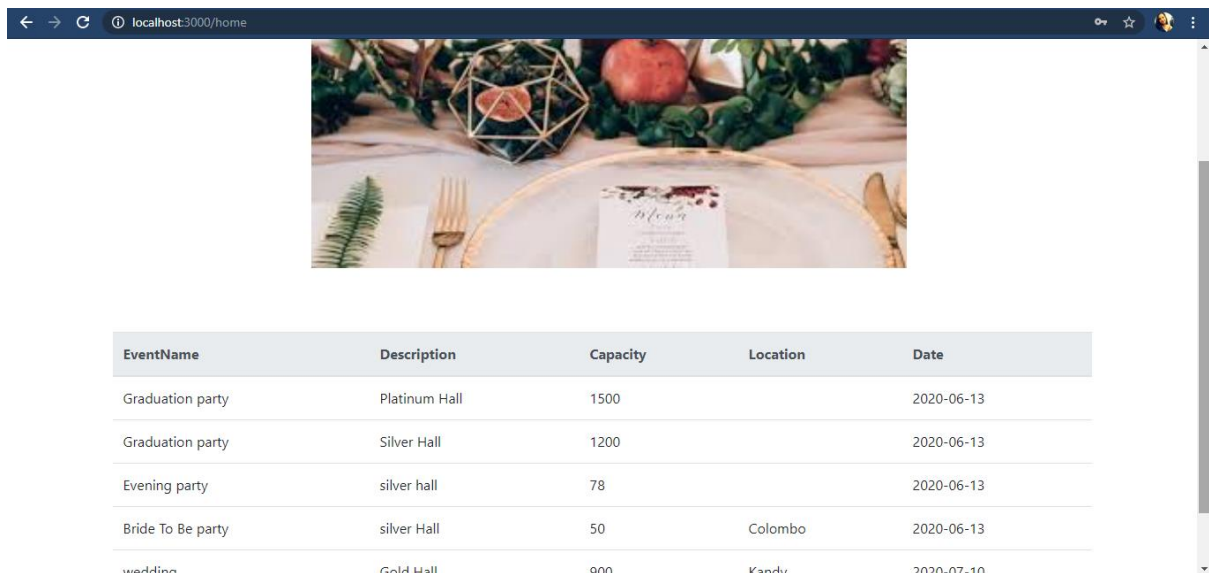
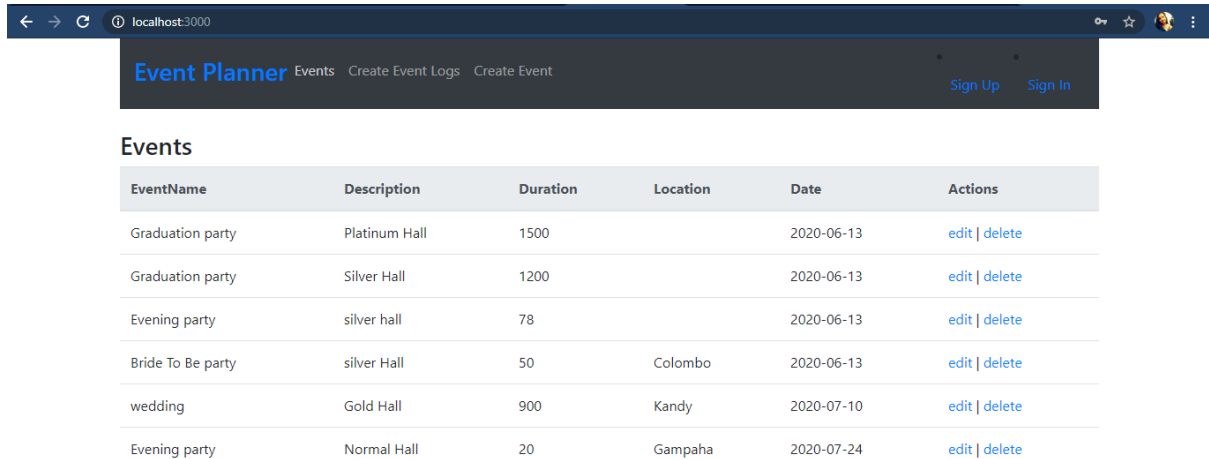


Figure : Details about the Events

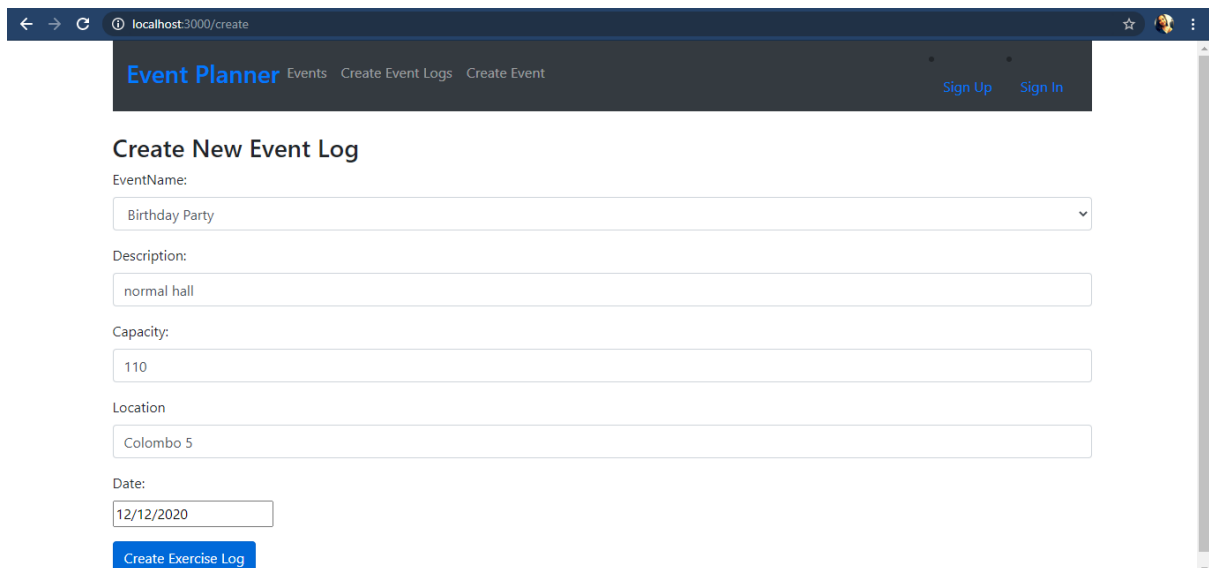
Event

The screenshot shows a web browser at localhost:3000 displaying the 'Event Planner' application. The navigation bar includes 'Events', 'Create Event Logs', and 'Create Event', along with 'Sign Up' and 'Sign In' buttons. The main content area is titled 'Events' and contains a table with the following data:

EventName	Description	Duration	Location	Date	Actions
Graduation party	Platinum Hall	1500		2020-06-13	edit delete
Graduation party	Silver Hall	1200		2020-06-13	edit delete
Evening party	silver hall	78		2020-06-13	edit delete
Bride To Be party	silver Hall	50	Colombo	2020-06-13	edit delete
wedding	Gold Hall	900	Kandy	2020-07-10	edit delete
Evening party	Normal Hall	20	Gampaha	2020-07-24	edit delete

Figure – Edit and delete events

This interface for admin . admin has a right to edit or delete events details.

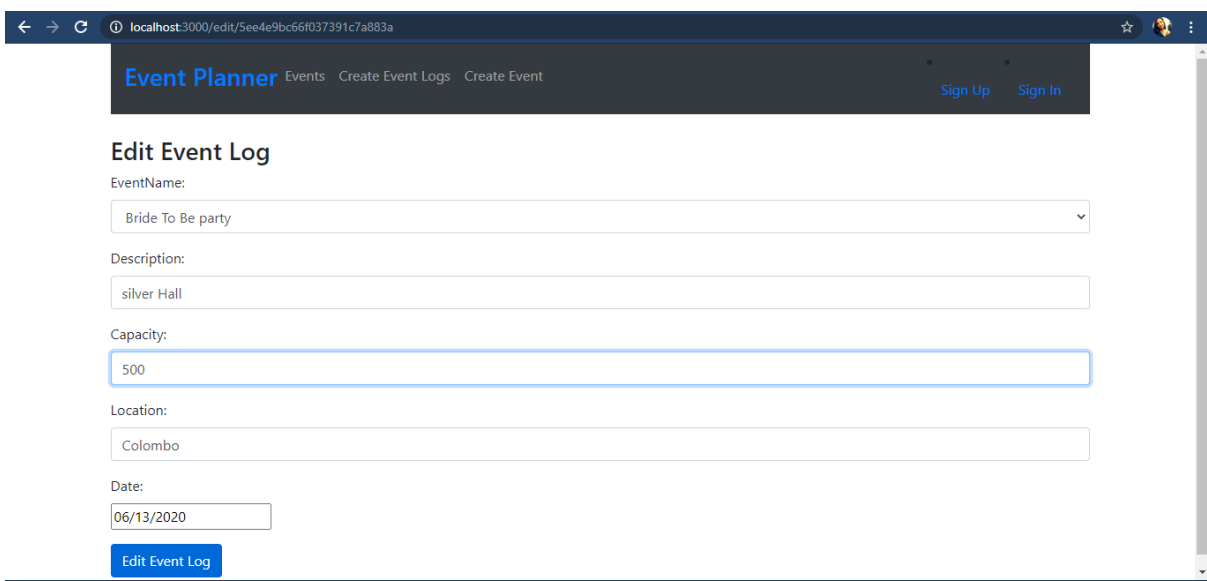
Create Event

The screenshot shows the 'Create New Event Log' form in the Event Planner application. The form includes the following fields and a submit button:

- EventName:** A dropdown menu with 'Birthday Party' selected.
- Description:** A text input field containing 'normal hall'.
- Capacity:** A text input field containing '110'.
- Location:** A text input field containing 'Colombo 5'.
- Date:** A text input field containing '12/12/2020'.
- Create Exercise Log:** A blue button to submit the form.

Figure – Create Event

Admin can create a event by adding details to these fields.

Edit Event

The screenshot shows a web browser window with the URL `localhost:3000/edit/See4e9bc66f037391c7a883a`. The page title is "Event Planner" and the navigation bar includes "Events", "Create Event Logs", and "Create Event". There are "Sign Up" and "Sign In" links in the top right corner. The main content area is titled "Edit Event Log" and contains the following form fields:

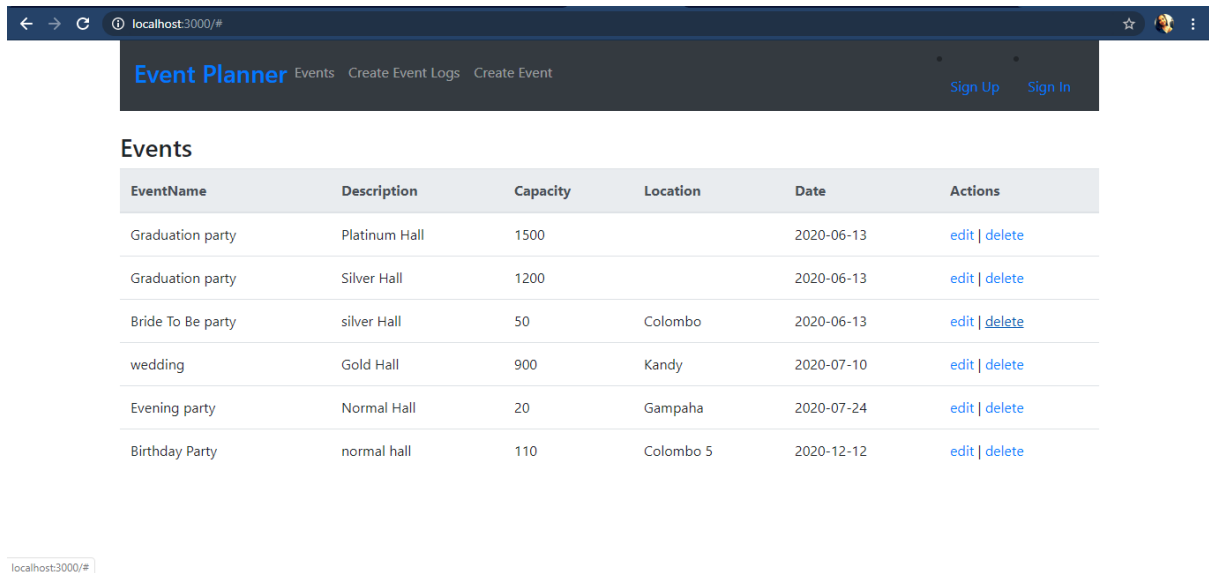
- EventName:** A dropdown menu with "Bride To Be party" selected.
- Description:** A text input field containing "silver Hall".
- Capacity:** A text input field containing "500".
- Location:** A text input field containing "Colombo".
- Date:** A text input field containing "06/13/2020".

At the bottom of the form is a blue button labeled "Edit Event Log".

Figure Edit Interface

Admin edit details about events.

Delete Event



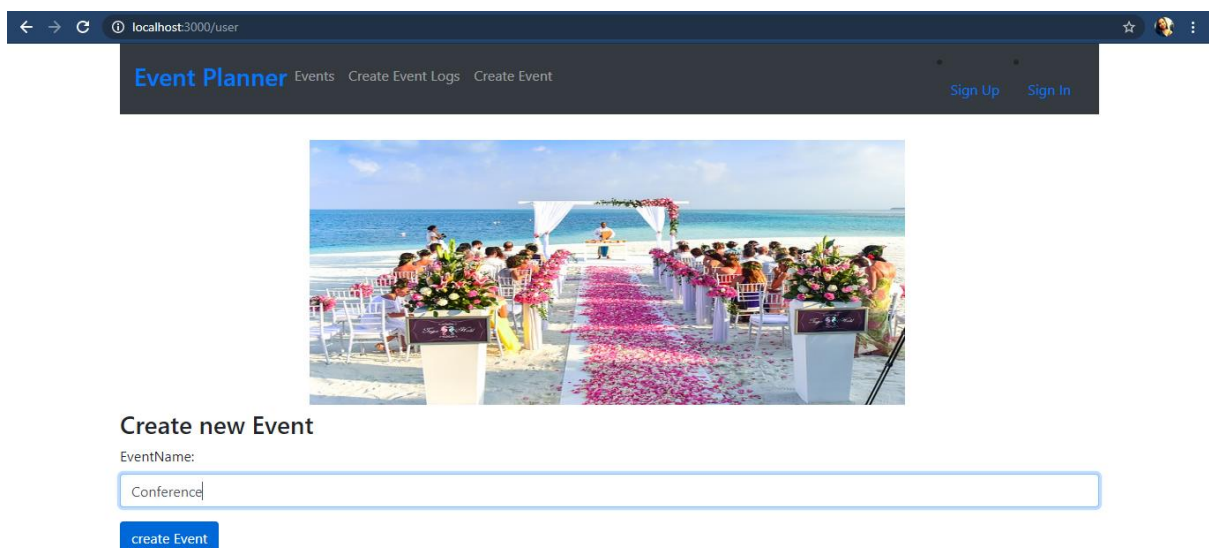
The screenshot shows a web browser at localhost:3000/#. The application header includes 'Event Planner' with links for 'Events', 'Create Event Logs', and 'Create Event'. There are 'Sign Up' and 'Sign In' buttons. Below the header, a table lists events with columns for EventName, Description, Capacity, Location, Date, and Actions. Each event has an 'edit | delete' link in the Actions column.

EventName	Description	Capacity	Location	Date	Actions
Graduation party	Platinum Hall	1500		2020-06-13	edit delete
Graduation party	Silver Hall	1200		2020-06-13	edit delete
Bride To Be party	silver Hall	50	Colombo	2020-06-13	edit delete
wedding	Gold Hall	900	Kandy	2020-07-10	edit delete
Evening party	Normal Hall	20	Gampaha	2020-07-24	edit delete
Birthday Party	normal hall	110	Colombo 5	2020-12-12	edit delete

Figure Delete Events

Admin can delete details about Events

Add Event Category



The screenshot shows the 'Create new Event' form in the Event Planner application. The browser address bar shows localhost:3000/user. The form has a text input field for 'EventName' with the value 'Conference' and a 'create Event' button. Above the form is a large image of a beach wedding ceremony.

Figure: Add Event Category

Admin can add Event category to system