

Internal Event Management System:

Introduction: In this management system, I have made an event booking calendar system so that each user can book their events for a particular day or a span of days

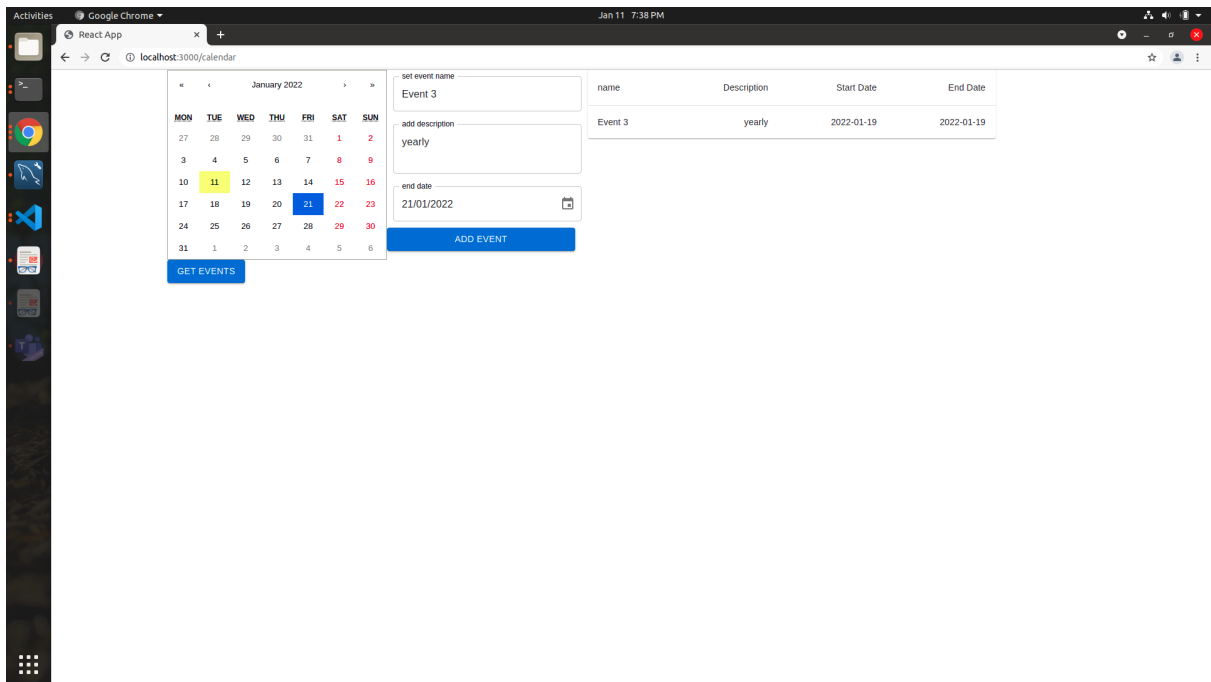
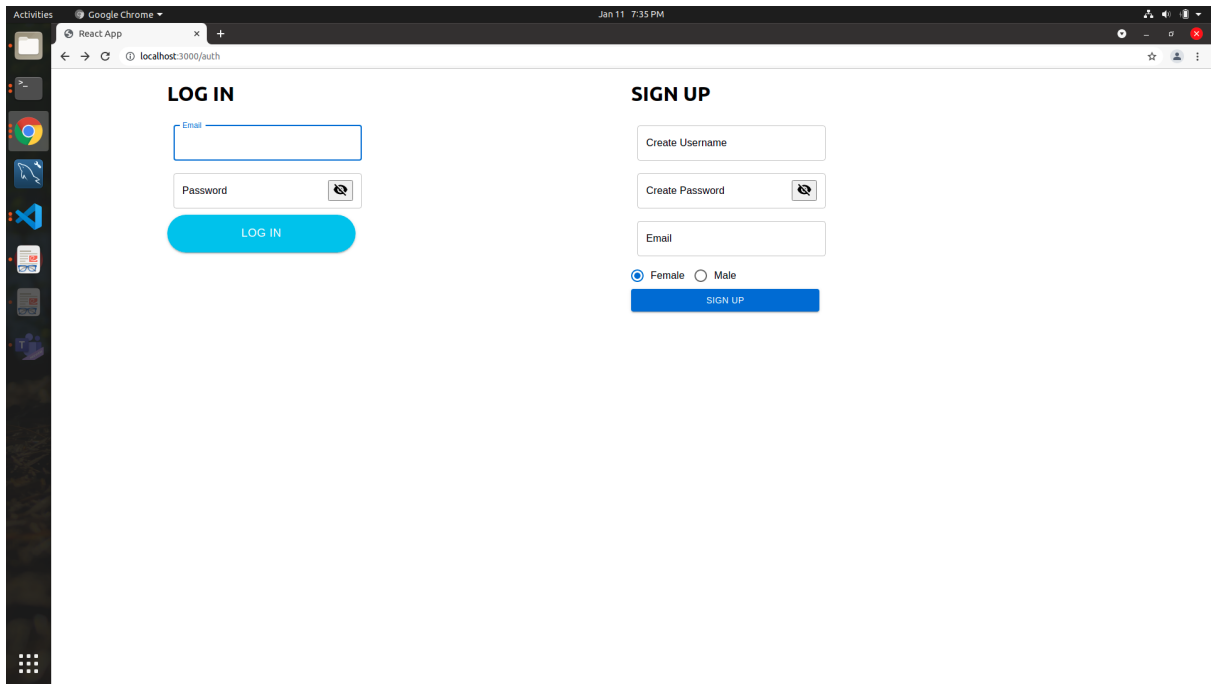
Prerequisites: To make this system, basic knowledge about programming and data structure in javascript, a basic introduction to react.js for frontend and node.js, and express js for backend is required. As well as knowledge of MySQL for the database management system is needed.

Details:

Database: The database has two tables for user and events. The user table contains columns as userID(uid), email, gender, and password. The events columns contain tables as eventID(id), userID(which is a foreign key from the user table), event name, description about the event, Start Date and End Date. The end date can be left blank while booking event . The program will take the end date as the start date.

Restful API: The stack used for the rest API is Node.js and Express.JS with MySQL database. Now a user can't sign up with the same email address and can only see his own events after he/she logs in. This is done to protect the privacy of the user. Only the host can see all the users and their event details but not the password.

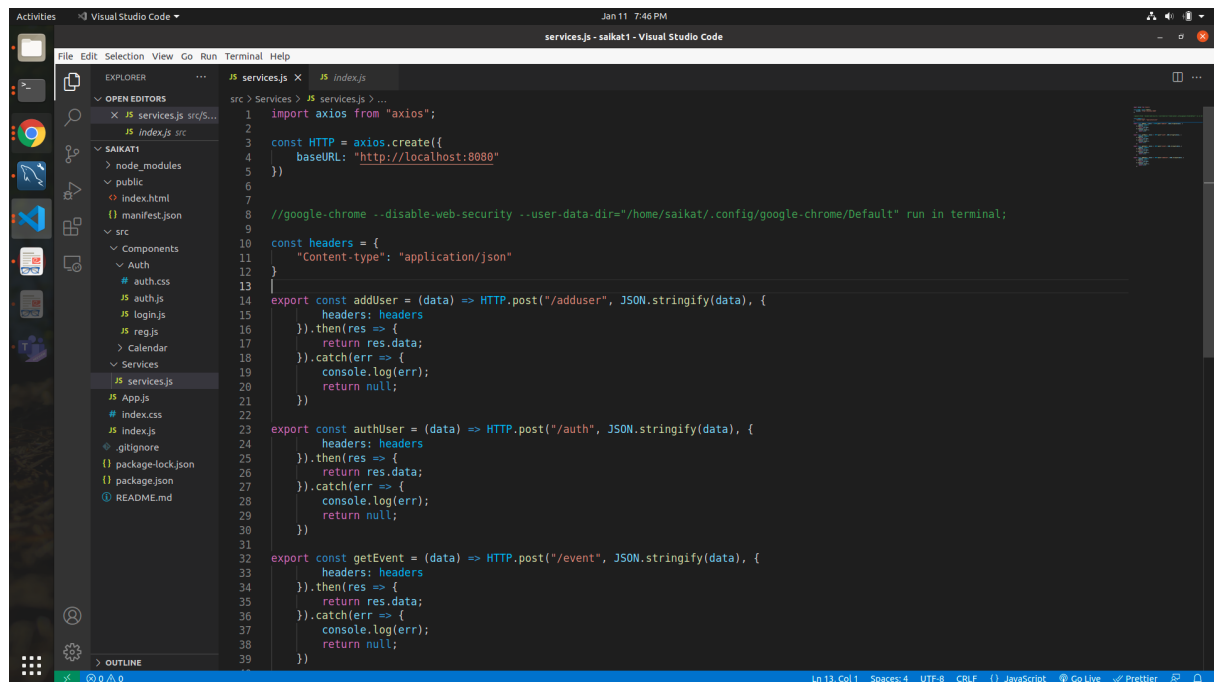
Frontend: The frontend contains two pages. Registration or Login page and event booking page. The frontend is created using react.js. Below we can see the two pages



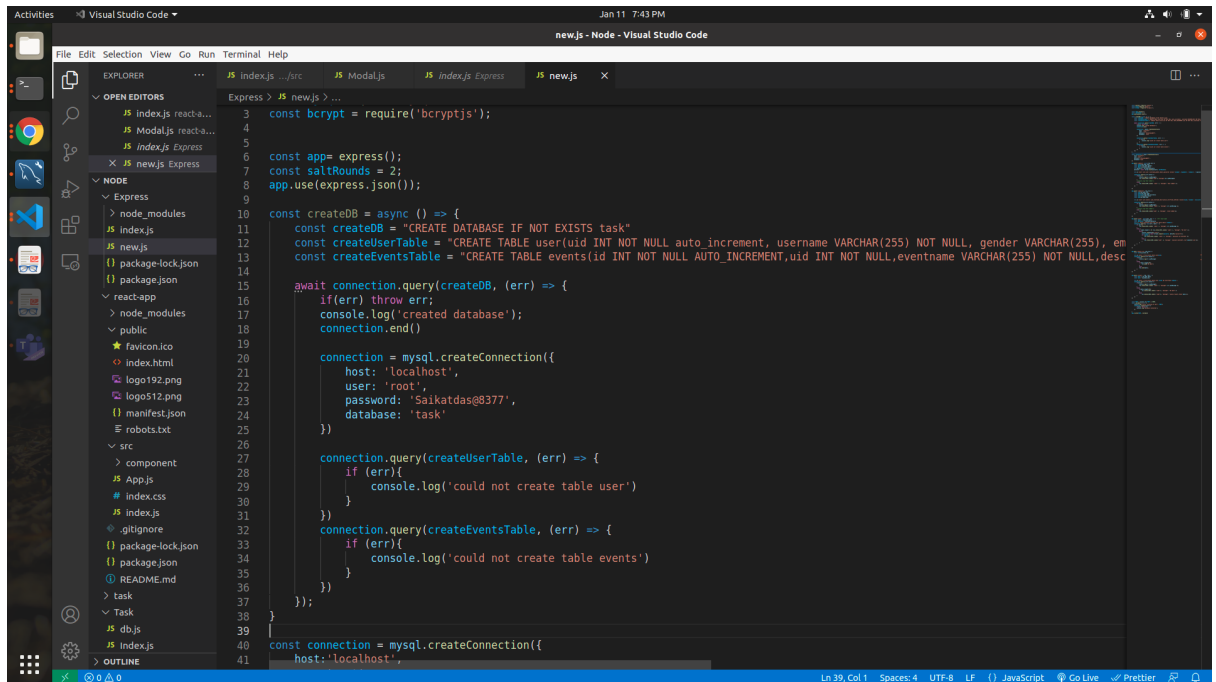
Setup: Now let's come to set up.

1. Disable your google chrome web security or you can't run both files in your own computer.

Run : `google-chrome --disable-web-security --user-data-dir="dir of your google chrome app" run in terminal`



2. In for rest api(can find in folder rest api), in file new.js change the details about your database.



```
const bcrypt = require('bcryptjs');
const app = express();
const saltRounds = 2;
app.use(express.json());

const createDB = async () => {
  const createDB = "CREATE DATABASE IF NOT EXISTS task";
  const createUserTable = "CREATE TABLE user(uid INT NOT NULL auto_increment, username VARCHAR(255) NOT NULL, gender VARCHAR(255), email VARCHAR(255) NOT NULL)";
  const createEventsTable = "CREATE TABLE events(id INT NOT NULL AUTO_INCREMENT, uid INT NOT NULL, eventname VARCHAR(255) NOT NULL, description VARCHAR(255) NOT NULL)";

  await connection.query(createDB, (err) => {
    if (err) throw err;
    console.log('created database');
    connection.end();
  });

  connection = mysql.createConnection({
    host: 'localhost',
    user: 'root',
    password: 'Saikatdas@8377',
    database: 'task'
  });

  connection.query(createUserTable, (err) => {
    if (err) {
      console.log('could not create table user');
    }
  });

  connection.query(createEventsTable, (err) => {
    if (err) {
      console.log('could not create table events');
    }
  });
});

const connection = mysql.createConnection({
  host: 'localhost',
```

You can see the function *await connection*, change the password (normally it is null or "" but if you have a password for your MySQL) and the database name(if you want a different database).

3. Finally,first, run *npm install* in both folders as due to cloning node modules may not be loaded then run *npm start* in saikat1 folder and *nodemon new.js* in rest api folder. When the web opens go to *localhost: PORT/auth*.

Enjoy!!