```
<nav aria-label="breadcrumb">
  <ol class="breadcrumb">
    <li class="breadcrumb-item"><a href="/">Home</a></li>
    <li class="breadcrumb-item active" aria-current="page">New User</li>
  </ol>
</nav>

<% if(alert) { %>
  <div class="alert alert-success alert-dismissible fade show" role="alert">
    <%= alert %>
    <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
  </div>
<% } %>

<form class="row g-3 needs-validation" method="POST" action="/adduser" novalidate>
  <%- include('partials/user-form.ejs', {row: {first_name: '', last_name: '', phone: '', email: '', comments: ''}}) %>
</form><nav aria-label="breadcrumb">
  <ol class="breadcrumb">
    <li class="breadcrumb-item"><a href="/">Home</a></li>
    <li class="breadcrumb-item active" aria-current="page">Edit User</li>
  </ol>
</nav>

<% if (alert) { %>
  <div class="alert alert-success alert-dismissible fade show" role="alert">
    <%= alert %>
    <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
  </div>
<% } %>

<% rows.forEach(function(row) { %>
<form class="row g-3 needs-validation" method="POST" action="/edituser/<%=row.id%>" novalidate>
  <%- include('partials/user-form.ejs', {row: row}) %>
</form>
<% }) %>
<% if (removedUser) { %>
<div class="alert alert-success alert-dismissible fade show" role="alert">
  User has been removed.
  <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
</div>
<% } %>

<div class="row">
  <div class="col-6">
    <h1>Users</h1>
  </div>
  <div class="col-6 d-flex justify-content-end">
    <a href="/adduser" type="button" class="btn btn-primary align-self-center">+ add user</a>
  </div>
</div>

<table class="table table-bordered">
  <thead class="thead-dark">
    <tr>
      <th scope="col">#</th>
      <th scope="col">First Name</th>
      <th scope="col">Last Name</th>
      <th scope="col">Email</th>
      <th scope="col">Phone</th>
      <th scope="col" class="text-end">Action</th>
    </tr>
  </thead>
  <tbody>

    <% rows.forEach(function(row) { %>

    <% if (row.status == "active") { %>
    <tr style="background-color: #90c090;">
    <% } else { %>
    <tr>
    <% } %>

      <th scope="row"><%row.id%></th>
      <td><%=row.first_name%></td>
      <td><%=row.last_name%></td>
      <td><%=row.email%></td>
      <td><%=row.phone%></td>

      <td class="text-end">

        <% if (row.status == "none") { %>
        <form class="g-3 needs-validation" method="POST" action="/activateuser/<%=row.id%>" novalidate>
          <input type="submit" class="btn btn-light btn-small" style="width: 10em !important;" value="Activate User">
        <% } else { %>
        <form class="g-3 needs-validation" method="POST" action="/deactivateuser/<%=row.id%>" novalidate>
          <input type="submit" class="btn btn-light btn-small" style="width: 10em !important;" value="Deactivate User">
        <% } %>

          <a href="/viewuser/<%=row.id%>" type="button" class="btn btn-light btn-small"><i class="bi bi-eye"></i> View<
          <a href="/edituser/<%=row.id%>" type="button" class="btn btn-light btn-small"><i class="bi bi-pencil"></i>
            Edit</a>
          <a href="/<%=row.id%>" type="button" class="btn btn-light btn-small"><i class="bi bi-person-x"></i> Delete</a
        </form>
      </td>
    </tr>
    <% }); %>

  </tbody>
</table><nav aria-label="breadcrumb">
  <ol class="breadcrumb">
    <li class="breadcrumb-item"><a href="/">Home</a></li>
    <li class="breadcrumb-item active" aria-current="page">View User</li>
  </ol>
</nav>

<% if (alert) { %>
  <div class="alert alert-success alert-dismissible fade show" role="alert">
    <%= alert %>
    <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
  </div>
<% } %>

<div class="view-user p-5">
  <% rows.forEach(function(row) { %>
  <div class="row mb-5">
    <div class="col text-center">
      <h3><%=row.first_name%> <%=row.last_name%></h3>
    </div>
  </div>
  <div class="row">
```

```
    <div class="col">
      <table class="table">
        <thead>
          <tr>
            <th scope="col">First Name</th>
            <th scope="col">Last Name</th>
            <th scope="col">Email</th>
            <th scope="col">Phone</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <th scope="row"><%=row.first_name%></th>
            <td><%=row.last_name%></td>
            <td><%=row.email%></td>
            <td><%=row.phone%></td>
          </tr>
        </tbody>
      </table>
    </div>
    <div class="row">
      <div class="col">
        <b>Comments</b>
      </div>
    </div>
    <div class="row">
      <div class="col">
        <%=row.comments%>
      </div>
    </div>
  </div>
  <% }) %>
</div><!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>User Management System</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1" crossorigin="an
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.3.0/font/bootstrap-icons.css">
</head>

<body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <div class="container-fluid">
      <a class="navbar-brand" href="/">User Management System</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedConte
        aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav me-auto mb-2 mb-lg-0">
          <li class="nav-item">
            <a class="nav-link active" aria-current="page" href="/">Home</a>
          </li>
        </ul>
        <form class="d-flex" method="POST" action="/" novalidate>
          <input class="form-control me-2" type="search" placeholder="Search" name="search" aria-label="Search">
          <button class="btn btn-outline-light" type="submit">Search</button>
        </form>

      </div>
    </div>
  </nav>

  <div class="container pt-5 pb-5">
    <%- body %>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-ygbV9kiqUc6oa4msXn9868pTtWMgiQaeYH7/t7LECLbyPA2x65Kgf80OJFdroafW"
    crossorigin="anonymous"></script>
</body>

</html>
  <div class="col-6">
    <div class="form-floating mb-3">
      <input type="text" class="form-control" id="floatingInput" value="<%=row.first_name%>" placeholder="First Nam
      <label for="floatingInput">First Name</label>
    </div>
  </div>
  <div class="col-6">
    <div class="form-floating mb-3">
      <input type="text" class="form-control" id="floatingInput" value="<%=row.last_name%>" placeholder="Last Name
      <label for="floatingInput">Last Name</label>
    </div>
  </div>
  <div class="col-6">
    <div class="form-floating mb-3">
      <input type="email" class="form-control" id="floatingInput" value="<%=row.email%>" placeholder="email@email
      <label for="floatingInput">Email</label>
    </div>
  </div>
  <div class="col-6">
    <div class="form-floating mb-3">
      <input type="text" class="form-control" id="floatingInput" value="<%=row.phone%>" placeholder="Phone" name
      <label for="floatingInput">Phone</label>
    </div>
  </div>
  <div class="col-12">
    <div class="form-floating mb-3">
      <textarea class="form-control" placeholder="Leave a comment here" id="comments" name="comments" style="h
      <label for="comments">Comments</label>
    </div>
  </div>
  <div class="col-12 d-grid">
    <button class="btn btn-primary" type="submit">Submit</button>
  </div>const express = require('express')
const expressLayouts = require('express-ejs-layouts');
//const exphbs = require('express-handlebars');
//const bodyParser = require('body-parser'); // No longer Required
//const mysql = require('mysql'); // Not required -> moved to userController

require('dotenv').config();

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

// Parsing middleware
// Parse application/x-www-form-urlencoded
// app.use(bodyParser.urlencoded({ extended: false }));
app.use(express.urlencoded({ extended: true })); // New

// Parse application/json
// app.use(bodyParser.json());
app.use(express.json()); // New

// Static Files
app.use(express.static('public'));

// Templating Engine
//const handlebars = exphbs.create({ extname: '.hbs', });
//app.engine('.hbs', handlebars.engine);
//app.set('view engine', '.hbs');
app.set('view engine', 'ejs');

// Layouts
app.use(expressLayouts);
app.set('layout', 'layouts/main.ejs');

// You don't need the connection here as we have it in userController
// let connection = mysql.createConnection({
//   host: process.env.DB_HOST,
//   user: process.env.DB_USER,
//   password: process.env.DB_PASS,
//   database: process.env.DB_NAME
// });

const routes = require('./routes/user');
app.use('/', routes);

app.listen(port, () => console.log(`Listening on port ${port}`));
const express = require('express');
const router = express.Router();
const userController = require('../controllers/userController');

// Routes
router.get('/', userController.view);
router.post('/', userController.find);
router.get('/adduser', userController.form);
router.post('/adduser', userController.create);
router.get('/edituser/:id', userController.edit);
router.post('/edituser/:id', userController.update);
router.get('/viewuser/:id', userController.viewall);
router.get('/:id',userController.delete);

router.post('/activateuser/:id', userController.activateUser);
router.post('/deactivateuser/:id', userController.deactivateUser);

module.exports = router;// import user Model
const User = require('../models/userModel');

// View Users
exports.view = (req, res) => {
  User.viewActiveUsers((err, rows) => {
    if (!err) {
      let removedUser = req.query.removed;
      res.render('home', { rows, removedUser: removedUser });
    } else {
      console.log(err);
    }
    console.log('The data from user table: \n', rows);
  });
}

// Find User by Search
exports.find = (req, res) => {
  let searchTerm = req.body.search;
  User.findUser(searchTerm, (err, rows) => {
    if (!err) {
      res.render('home', { rows, removedUser: null });
    } else {
      console.log(err);
    }
    console.log('The data from user table: \n', rows);
  });
}

exports.form = (req, res) => {
  res.render('add-user', { alert: '' });
}

// Add new user
exports.create = (req, res) => {
  const { first_name, last_name, email, phone, comments } = req.body;

  User.addUser({
      first_name: first_name,
      last_name: last_name,
      email: email,
      phone: phone,
      comments: comments
  }, (err, rows) => {
    if (!err) {
      res.render('add-user', { alert: 'User added successfully.' });
    } else {
      console.log(err);
    }
    console.log('The data from user table: \n', rows);
  });
}


// Edit user
exports.edit = (req, res) => {
  User.editUser(req.params.id, (err, rows) => {
    if (!err) {
      res.render('edit-user', { rows, alert: '' });
    } else {
      console.log(err);
    }
    console.log('The data from user table: \n', rows);
  });
}
```

```
// Update User
exports.update = (req, res) => {
  const { first_name, last_name, email, phone, comments } = req.body;
  User.updateUser(req.params.id, {
      first_name: first_name,
      last_name: last_name,
      email: email,
      phone: phone,
      comments: comments
  }, (err, rows) => {
    if (!err) {
      User.viewUsers(req.params.id, (err, rows) => {

        if (!err) {
          res.render('edit-user', { rows, alert: `${first_name} has been updated.` });
        } else {
          console.log(err);
        }
        console.log('The data from user table: \n', rows);
      });
    } else {
      console.log(err);
    }
    console.log('The data from user table: \n', rows);
  });
}

// Delete User
exports.delete = (req, res) => {

  // Delete a record

  // connection.query('DELETE FROM user WHERE id = ?', [req.params.id], (err, rows) => {

  //   if(!err) {
  //     res.redirect('/');
  //   } else {
  //     console.log(err);
  //   }
  //   console.log('The data from user table: \n', rows);

  // });

  // Hide a record

  User.deleteUser(req.params.id, (err, rows) => {
    if (!err) {
      let removedUser = encodeURIComponent('User successeflly removed.');
      res.redirect('/?removed=' + removedUser);
    } else {
      console.log(err);
    }
    console.log('The data from beer table are: \n', rows);
  });

}

// View Users
exports.viewall = (req, res) => {
  User.viewUsers(req.params.id, (err, rows) => {
    if (!err) {
      res.render('view-user', { rows, alert: '' });
    } else {
      console.log(err);
    }
    console.log('The data from user table: \n', rows);
  });
}

// Activate User
exports.activateUser = (req, res) => {
  User.activateUser(req.params.id, (err, rows) => {
    if (!err) {
      User.viewActiveUsers((err, rows) => {
        if (!err) {
          res.render('home', { rows, removedUser: null });
        } else {
          console.log(err);
        }

        console.log('The data from user table: \n', rows);
      });
    } else {
      console.log(err);
    }
    console.log('The data from user table: \n', rows);
  });
}

// Deactivate User
exports.deactivateUser = (req, res) => {
  User.deactivateUser(req.params.id, (err, rows) => {
    if (!err) {
      User.viewActiveUsers((err, rows) => {
        if (!err) {
          res.render('home', { rows, removedUser: null });
        } else {
          console.log(err);
        }

        console.log('The data from user table: \n', rows);
      });
    } else {
      console.log(err);
    }
    console.log('The data from user table: \n', rows);
  });
}const mysql = require('mysql');

class User {
    constructor() {
        this.connection = mysql.createConnection({
            host: process.env.DB_HOST,
            user: process.env.DB_USER,
            password: process.env.DB_PASS,
            database: process.env.DB_NAME
        });
    }
    // view active users
    // we now consider active users as users with stus none or status active, because we want to see both types of use
    viewActiveUsers(callback) {
        this.connection.query('SELECT * FROM user WHERE status = "active" OR status = "none"', (err, rows) => {

            if (err) callback(err, null);
            else callback(null, rows);
        });
    };

    // find user by search
    findUser(searchTerm, callback) {
        this.connection.query('SELECT * FROM user WHERE first_name LIKE ? OR last_name LIKE ?', ['%' + searchTerm + '

            if (err) callback(err, null);
            else callback(null, rows);
        });
    };

    // add user to database
    addUser(props, callback) {
        this.connection.query('INSERT INTO user SET first_name = ?, last_name = ?, email = ?, phone = ?, comments = ?', [
  props.comments], (err, rows) => {

            if (err) callback(err, null);
            else callback(null, rows);
        });
    };

    // edit user
    editUser(id, callback) {
        this.connection.query('SELECT * FROM user WHERE id = ?', [id], (err, rows) => {

            if (err) callback(err, null);
            else callback(null, rows);
        });
    };

    // update user
    updateUser(id, props, callback) {
        this.connection.query('UPDATE user SET first_name = ?, last_name = ?, email = ?, phone = ?, comments = ? WHER
 s.phone, props.comments, id], (err, rows) => {

            if (err) callback(err, null);
            else callback(null, rows);
        });
    }

    // delete user
    deleteUser(id, callback) {
        this.connection.query('UPDATE user SET status = ? WHERE id = ?', ['removed', id], (err, rows) => {

            if (err) callback(err, null);
            else callback(null, rows);
        });
    };

    // view users
    viewUsers(id, callback) {
        this.connection.query('SELECT * FROM user WHERE id = ?', [id], (err, rows) => {
            if (err) callback(err, null);
            else callback(null, rows);
        });
    };

    // activate user
    activateUser(id, callback) {
        this.connection.query('UPDATE user SET status = ? WHERE id = ?', ['active', id], (err, rows) => {

            if (err) callback(err, null);
            else callback(null, rows);
        });
    }

    // deactivate user
    deactivateUser(id, callback) {
        this.connection.query('UPDATE user SET status = ? WHERE id = ?', ['none', id], (err, rows) => {

            if (err) callback(err, null);
            else callback(null, rows);
        });
    }
}

module.exports = new User();
```