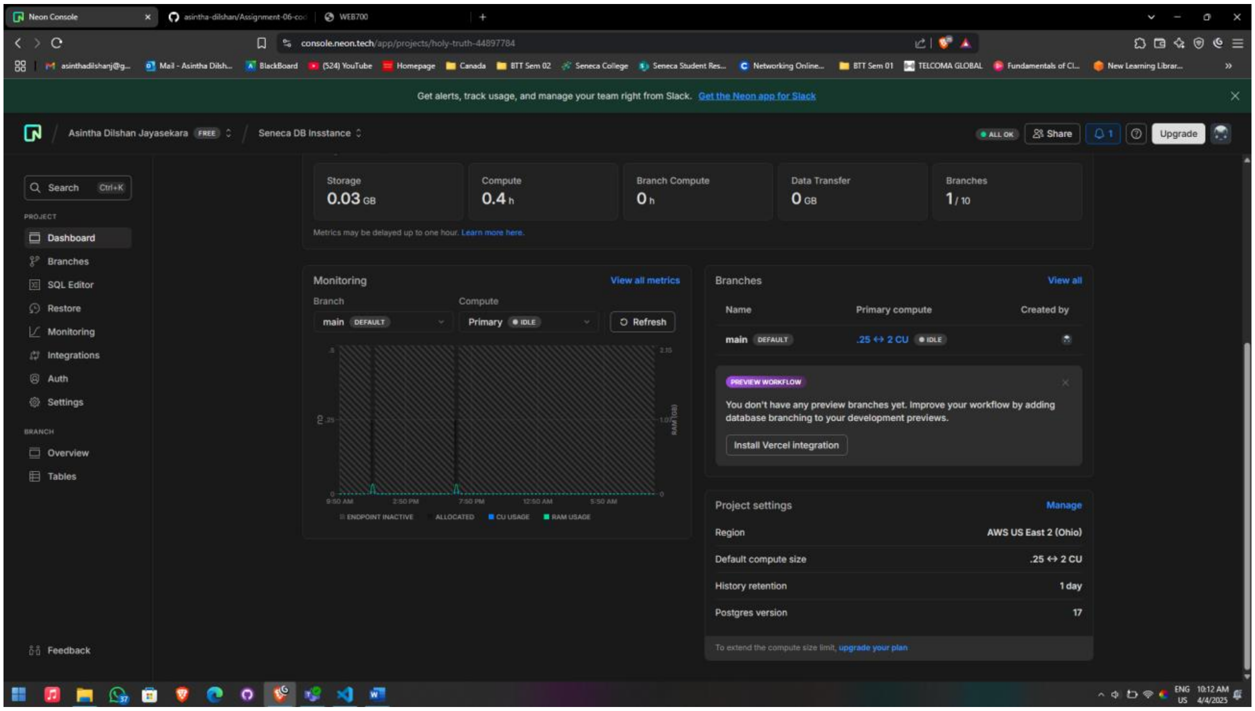
WEB700_Assignment6_Asintha Dilshan Jayasekara (170388235)

GiitHub Link: https://github.com/asintha-dilshan/Assignment-06-code.git

Vercel Link: https://assignment-06-code.vercel.app/

Neon.tech Databse



Get alerts, track usage, and manage your team right from Slack. Get the Neon app for Slack

Storage	Compute	Branch Compute	Data Transfer	Branches
0.03 GB	0.4 h	0 h	0 GB	1 / 10

Metrics may be delayed up to one hour. Learn more here.

Monitoring

Branch: main DEFAULT

Compute: Primary IDLE

Branches

Name	Primary compute	Created by
main DEFAULT	.25 ↔ 2 CU IDLE	

PREVIEW WORKFLOW

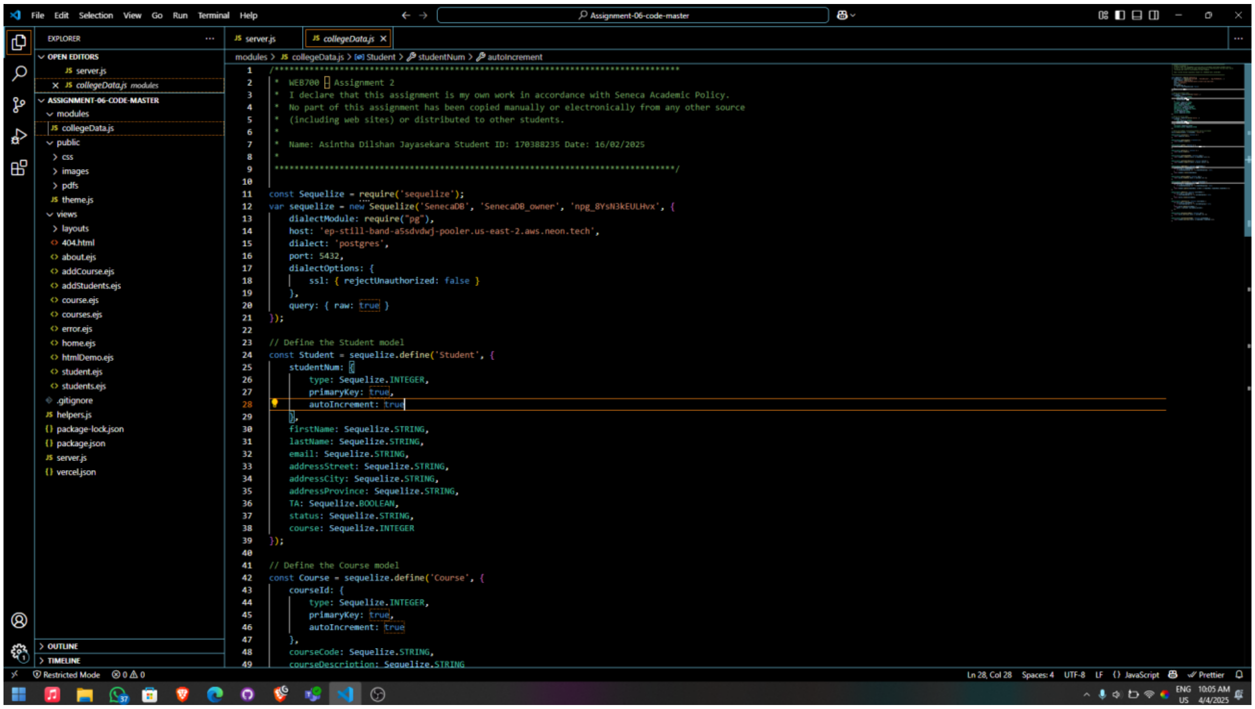
You don't have any preview branches yet. Improve your workflow by adding database branching to your development previews.

Project settings

Region	AWS US East 2 (Ohio)
Default compute size	.25 ↔ 2 CU
History retention	1 day
Postgres version	17

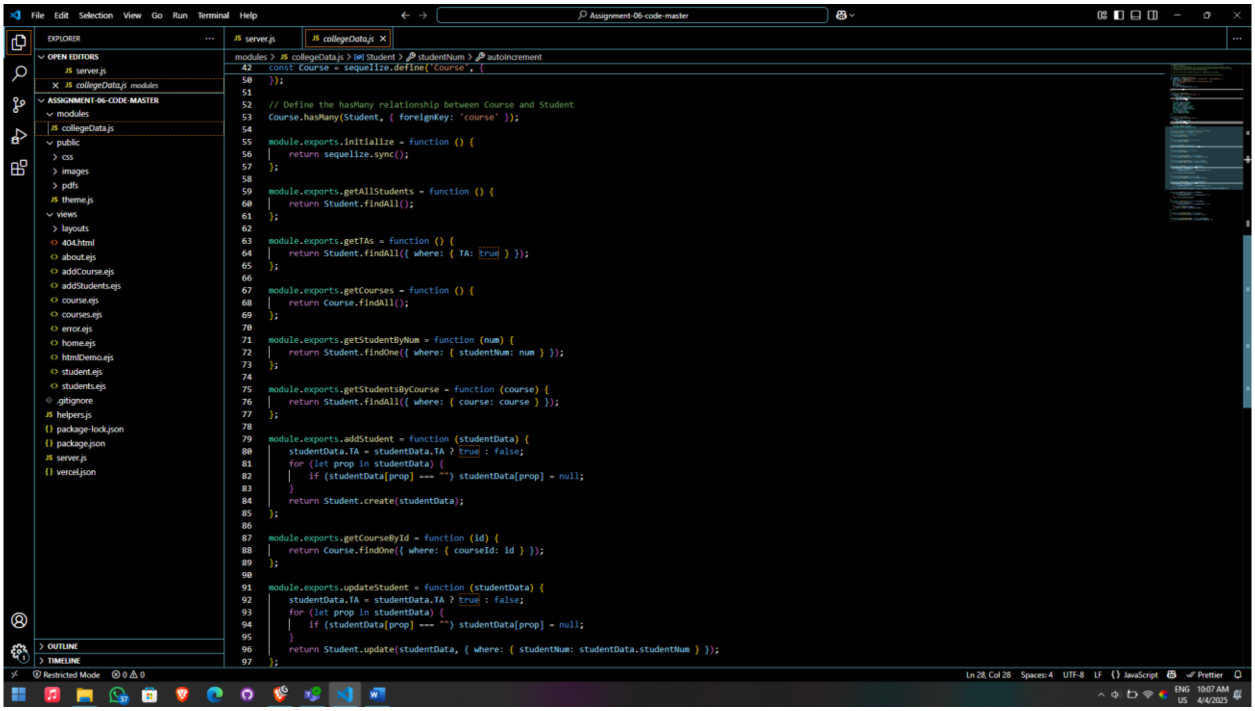
To extend the compute size limit, upgrade your plan

collegeData.js

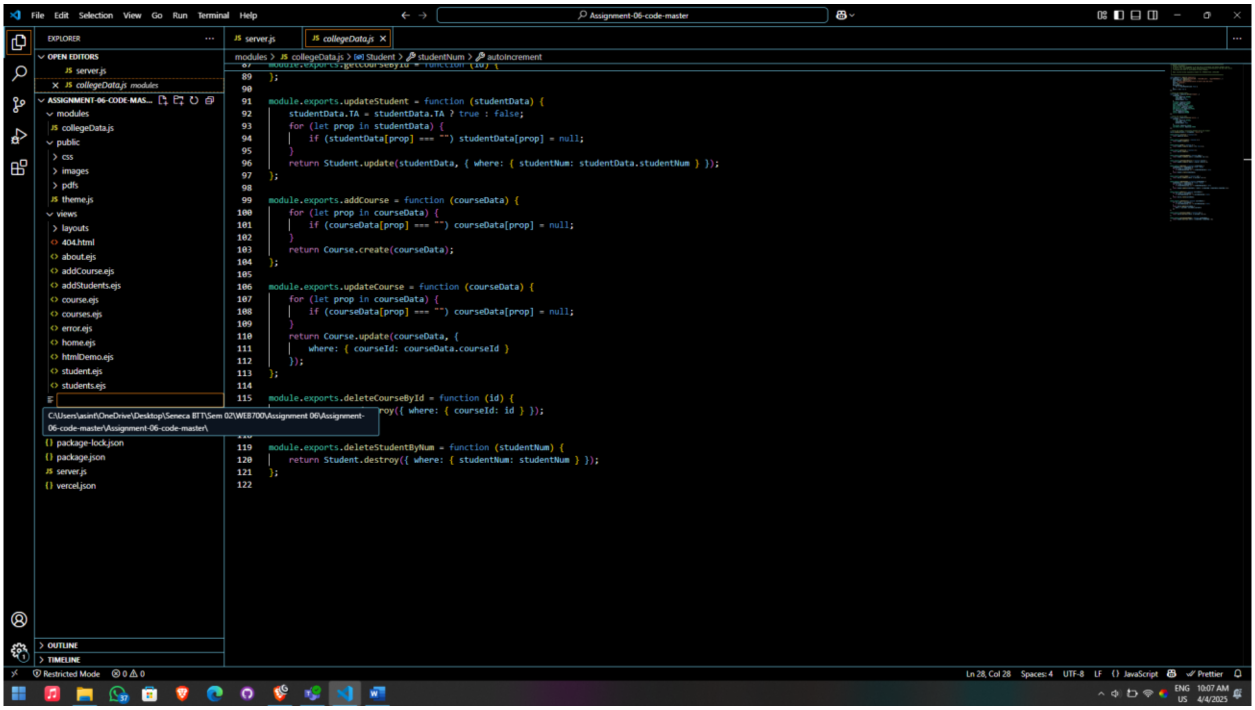


```
/*********************************************************************************
*  WEB700 – Assignment 2
*  I declare that this assignment is my own work in accordance with Seneca Academic Policy.
*  No part of this assignment has been copied manually or electronically from any other source
*  (including web sites) or distributed to other students.
*
*  Name: Asintha Dilshan Jayasekara Student ID: 170388235 Date: 16/02/2025
*
********************************************************************************/

const Sequelize = require('sequelize');
var sequelize = new Sequelize('SenecaDB', 'SenecaDB_owner', 'npg_8YsN3kEULHvx', {
    dialectModule: require("pg"),
    host: 'ep-still-band-a5sdvdwj-pooler.us-east-2.aws.neon.tech',
    dialect: 'postgres',
    port: 5432,
    dialectOptions: {
        ssl: { rejectUnauthorized: false }
    },
    query: { raw: true }
});

// Define the Student model
const Student = sequelize.define('Student', {
    studentNum: {
        type: Sequelize.INTEGER,
        primaryKey: true,
        autoIncrement: true
    },
    firstName: Sequelize.STRING,
    lastName: Sequelize.STRING,
    email: Sequelize.STRING,
    addressStreet: Sequelize.STRING,
    addressCity: Sequelize.STRING,
    addressProvince: Sequelize.STRING,
    TA: Sequelize.BOOLEAN,
    status: Sequelize.STRING,
    course: Sequelize.INTEGER
});

// Define the Course model
const Course = sequelize.define('Course', {
    courseId: {
        type: Sequelize.INTEGER,
        primaryKey: true,
        autoIncrement: true
    },
    courseCode: Sequelize.STRING,
    courseDescription: Sequelize.STRING
```

WEB700_Assignment6_Asintha Dilshan Jayasekara (170388235)



```javascript
const Course = sequelize.define('Course', {
});

// Define the hasMany relationship between Course and Student
Course.hasMany(Student, { foreignKey: 'course' });

module.exports.initialize = function () {
    return sequelize.sync();
};

module.exports.getAllStudents = function () {
    return Student.findAll();
};

module.exports.getTAs = function () {
    return Student.findAll({ where: { TA: true } });
};

module.exports.getCourses = function () {
    return Course.findAll();
};

module.exports.getStudentByNum = function (num) {
    return Student.findOne({ where: { studentNum: num } });
};

module.exports.getStudentsByCourse = function (course) {
    return Student.findAll({ where: { course: course } });
};

module.exports.addStudent = function (studentData) {
    studentData.TA = studentData.TA ? true : false;
    for (let prop in studentData) {
        if (studentData[prop] === "") studentData[prop] = null;
    }
    return Student.create(studentData);
};

module.exports.getCourseById = function (id) {
    return Course.findOne({ where: { courseId: id } });
};

module.exports.updateStudent = function (studentData) {
    studentData.TA = studentData.TA ? true : false;
    for (let prop in studentData) {
        if (studentData[prop] === "") studentData[prop] = null;
    }
    return Student.update(studentData, { where: { studentNum: studentData.studentNum } });
};
```

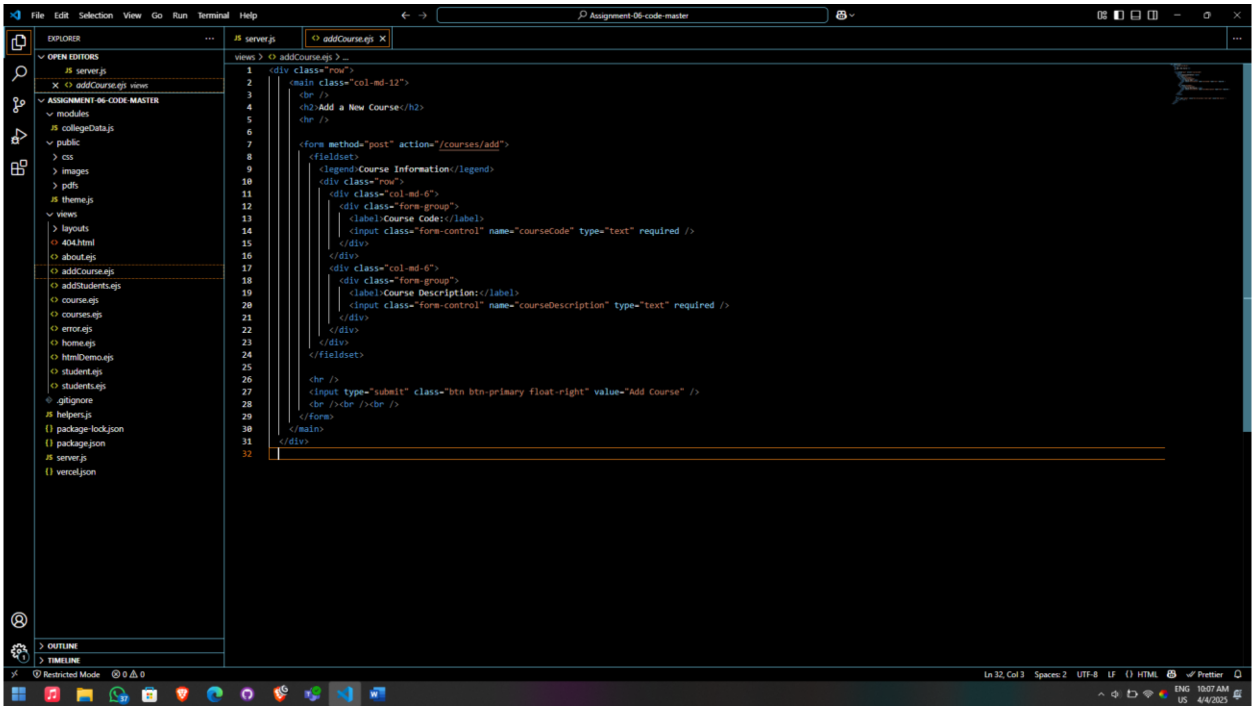


C:\Users\asint\OneDrive\Desktop\Seneca BTT\Sem 02\WEB700\Assignment 06\Assignment-06-code-master\Assignment-06-code-master\

```javascript
module.exports.getCourseById = function (id) {
};

module.exports.updateStudent = function (studentData) {
    studentData.TA = studentData.TA ? true : false;
    for (let prop in studentData) {
        if (studentData[prop] === "") studentData[prop] = null;
    }
    return Student.update(studentData, { where: { studentNum: studentData.studentNum } });
};

module.exports.addCourse = function (courseData) {
    for (let prop in courseData) {
        if (courseData[prop] === "") courseData[prop] = null;
    }
    return Course.create(courseData);
};

module.exports.updateCourse = function (courseData) {
    for (let prop in courseData) {
        if (courseData[prop] === "") courseData[prop] = null;
    }
    return Course.update(courseData, {
        where: { courseId: courseData.courseId }
    });
};

module.exports.deleteCourseById = function (id) {
                          roy({ where: { courseId: id } });

module.exports.deleteStudentByNum = function (studentNum) {
    return Student.destroy({ where: { studentNum: studentNum } });
};
```

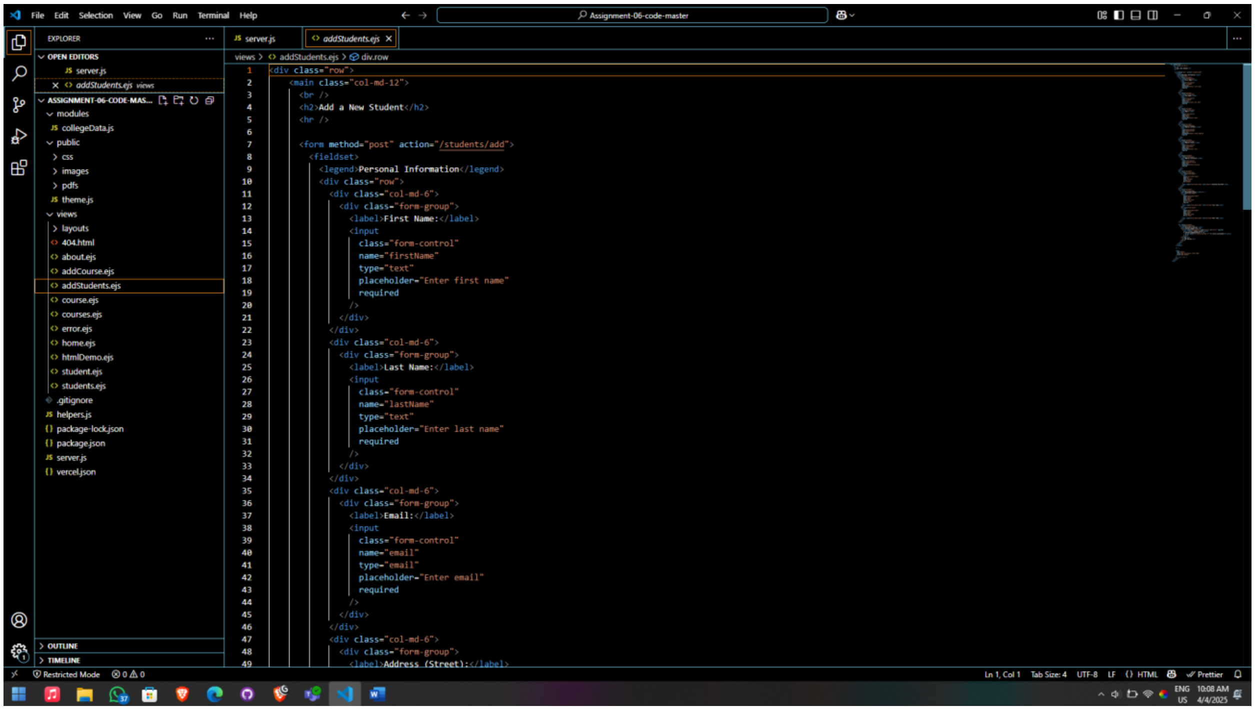
WEB700_Assignment6_Asintha Dilshan Jayasekara (170388235)

addCourse.ejs



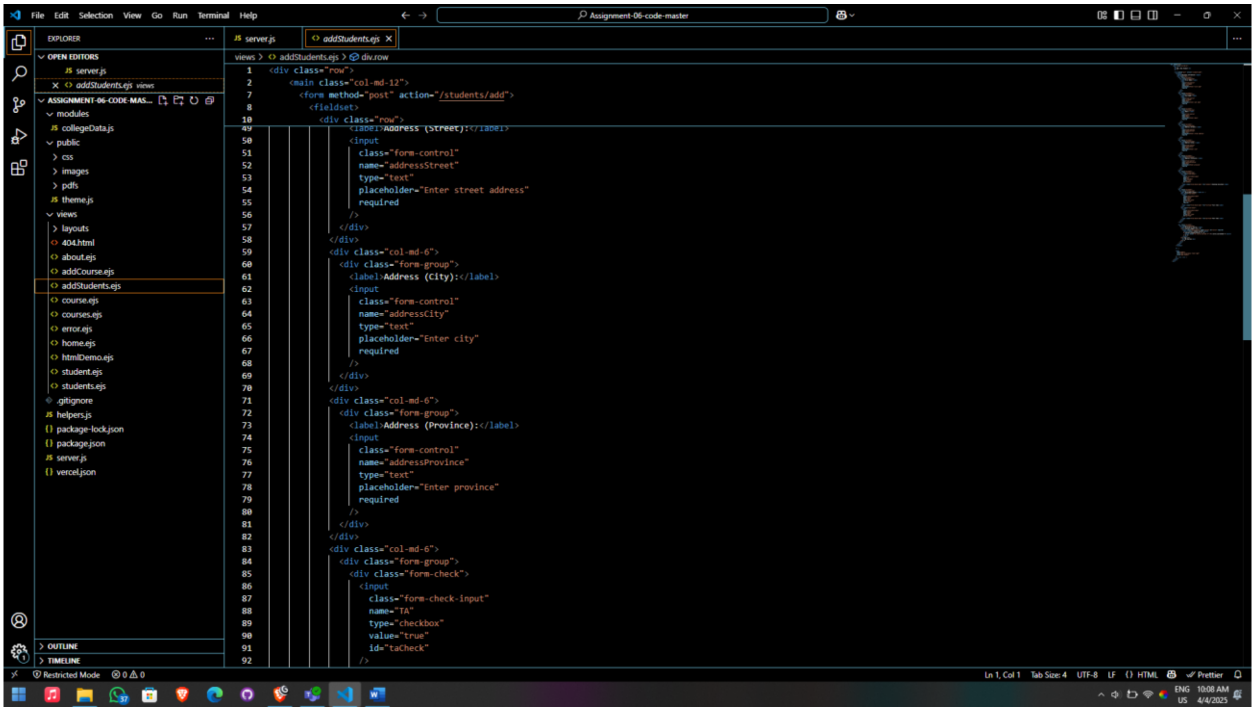
```
<div class="row">
    <main class="col-md-12">
      <br />
      <h2>Add a New Course</h2>
      <hr />

      <form method="post" action="/courses/add">
        <fieldset>
          <legend>Course Information</legend>
          <div class="row">
            <div class="col-md-6">
              <div class="form-group">
                <label>Course Code:</label>
                <input class="form-control" name="courseCode" type="text" required />
              </div>
            </div>
            <div class="col-md-6">
              <div class="form-group">
                <label>Course Description:</label>
                <input class="form-control" name="courseDescription" type="text" required />
              </div>
            </div>
          </div>
        </fieldset>

        <hr />
        <input type="submit" class="btn btn-primary float-right" value="Add Course" />
        <br /><br /><br />
      </form>
    </main>
  </div>

```

addStudent.ejs

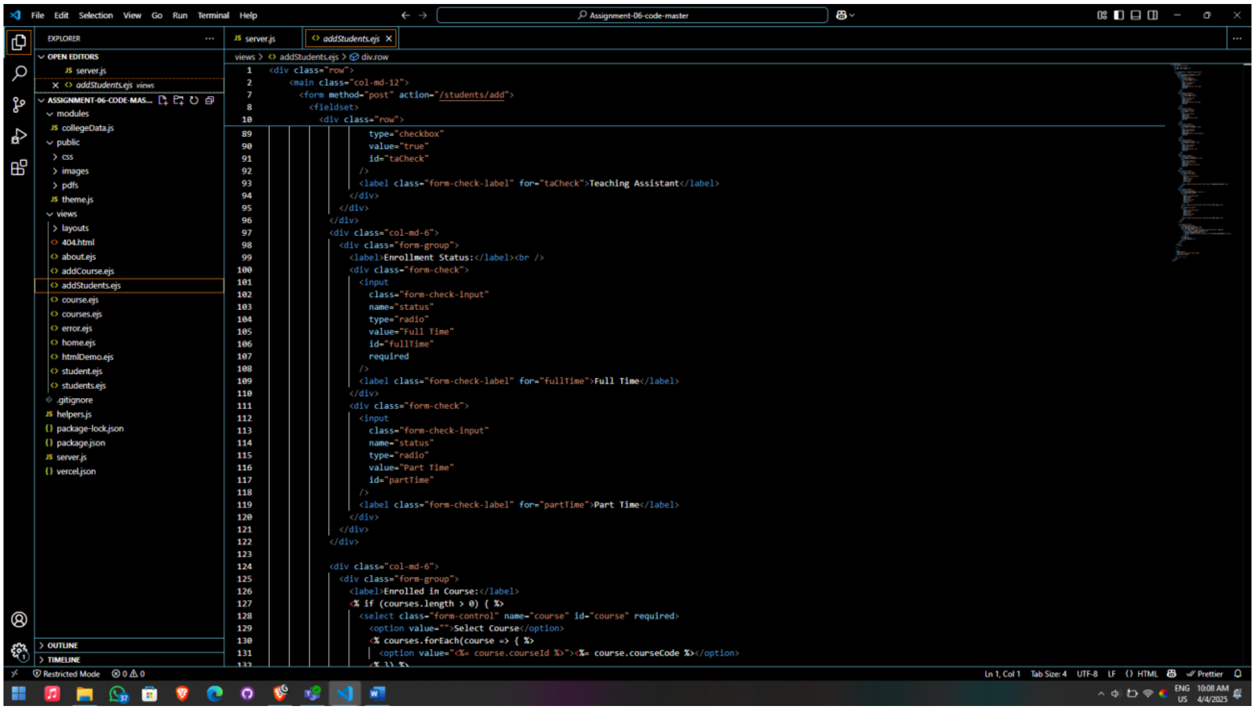


```
<div class="row">
    <main class="col-md-12">
      <br />
      <h2>Add a New Student</h2>
      <hr />

      <form method="post" action="/students/add">
        <fieldset>
          <legend>Personal Information</legend>
          <div class="row">
            <div class="col-md-6">
              <div class="form-group">
                <label>First Name:</label>
                <input
                  class="form-control"
                  name="firstName"
                  type="text"
                  placeholder="Enter first name"
                  required
                />
              </div>
            </div>
            <div class="col-md-6">
              <div class="form-group">
                <label>Last Name:</label>
                <input
                  class="form-control"
                  name="lastName"
                  type="text"
                  placeholder="Enter last name"
                  required
                />
              </div>
            </div>
            <div class="col-md-6">
              <div class="form-group">
                <label>Email:</label>
                <input
                  class="form-control"
                  name="email"
                  type="email"
                  placeholder="Enter email"
                  required
                />
              </div>
            </div>
            <div class="col-md-6">
              <div class="form-group">
                <label>Address (Street):</label>
```

WEB700_Assignment6_Asintha Dilshan Jayasekara (170388235)

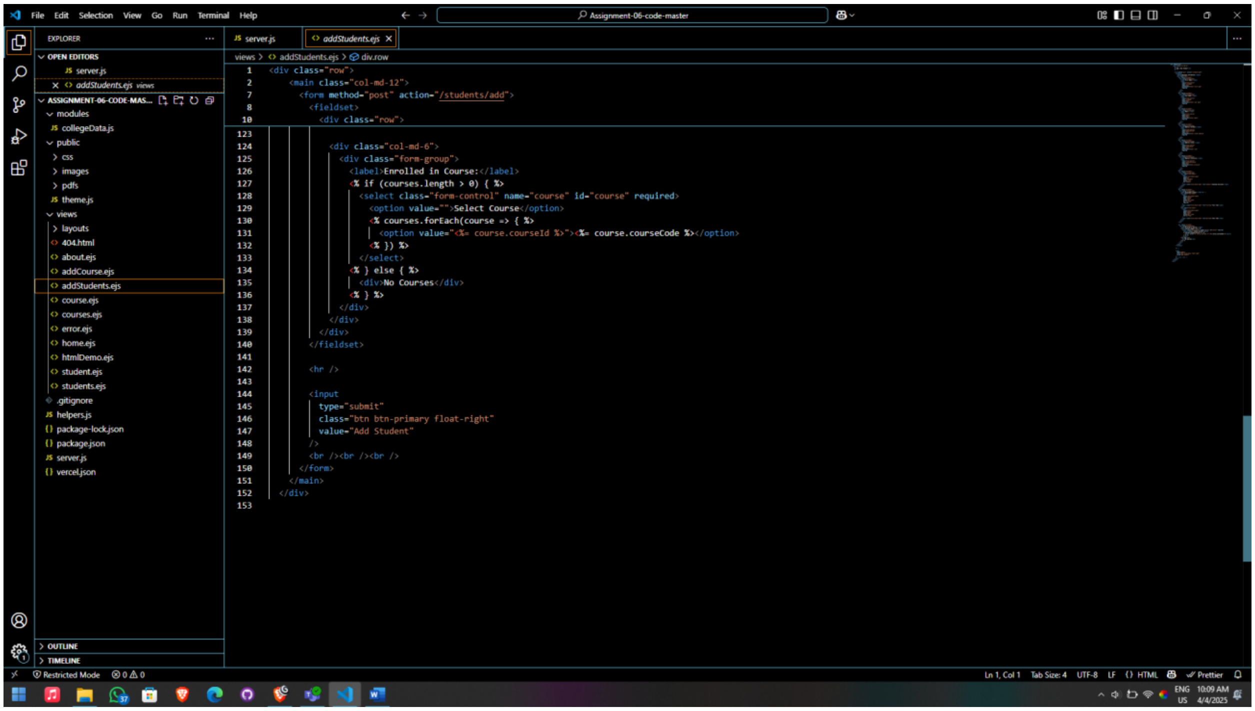


```
<div class="row">
    <main class="col-md-12">
      <form method="post" action="/students/add">
        <fieldset>
          <div class="row">
                <label>Address (Street):</label>
                <input
                  class="form-control"
                  name="addressStreet"
                  type="text"
                  placeholder="Enter street address"
                  required
                />
              </div>
            </div>
            <div class="col-md-6">
              <div class="form-group">
                <label>Address (City):</label>
                <input
                  class="form-control"
                  name="addressCity"
                  type="text"
                  placeholder="Enter city"
                  required
                />
              </div>
            </div>
            <div class="col-md-6">
              <div class="form-group">
                <label>Address (Province):</label>
                <input
                  class="form-control"
                  name="addressProvince"
                  type="text"
                  placeholder="Enter province"
                  required
                />
              </div>
            </div>
            <div class="col-md-6">
              <div class="form-group">
                <div class="form-check">
                  <input
                    class="form-check-input"
                    name="TA"
                    type="checkbox"
                    value="true"
                    id="taCheck"
                  />
```



```
<div class="row">
    <main class="col-md-12">
      <form method="post" action="/students/add">
        <fieldset>
          <div class="row">
                    type="checkbox"
                    value="true"
                    id="taCheck"
                  />
                  <label class="form-check-label" for="taCheck">Teaching Assistant</label>
                </div>
              </div>
            </div>
            <div class="col-md-6">
              <div class="form-group">
                <label>Enrollment Status:</label><br />
                <div class="form-check">
                  <input
                    class="form-check-input"
                    name="status"
                    type="radio"
                    value="Full Time"
                    id="fullTime"
                    required
                  />
                  <label class="form-check-label" for="fullTime">Full Time</label>
                </div>
                <div class="form-check">
                  <input
                    class="form-check-input"
                    name="status"
                    type="radio"
                    value="Part Time"
                    id="partTime"
                  />
                  <label class="form-check-label" for="partTime">Part Time</label>
                </div>
              </div>
            </div>

            <div class="col-md-6">
              <div class="form-group">
                <label>Enrolled in Course:</label>
                <% if (courses.length > 0) { %>
                  <select class="form-control" name="course" id="course" required>
                    <option value="">Select Course</option>
                    <% courses.forEach(course => { %>
                      <option value="<%= course.courseId %>"><%= course.courseCode %></option>
                    <% }) %>
```

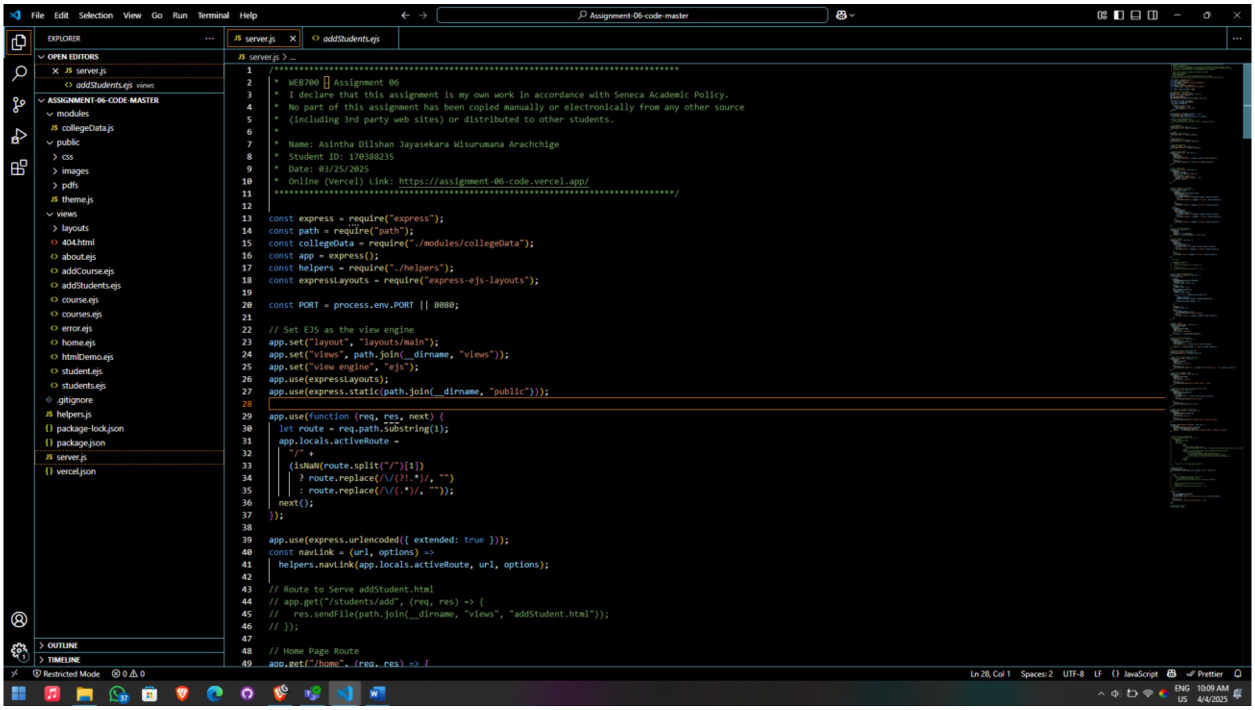
WEB700_Assignment6_Asintha Dilshan Jayasekara (170388235)



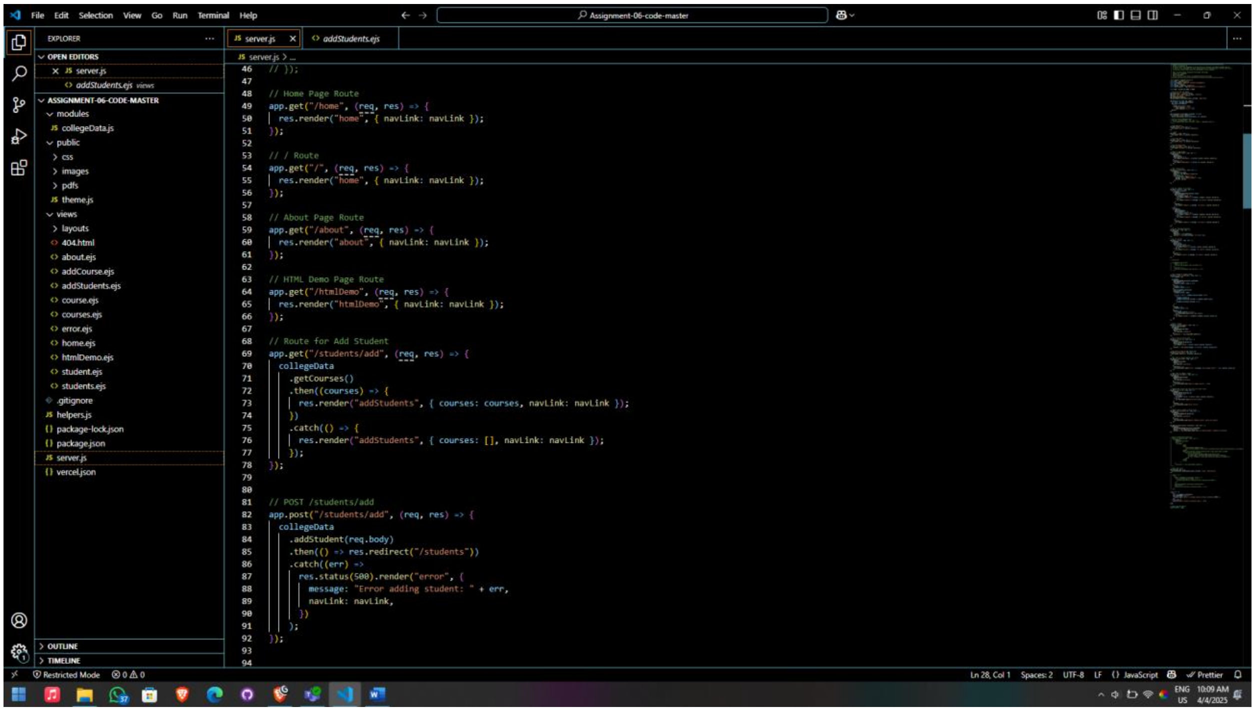
```
<div class="row">
  <main class="col-md-12">
    <form method="post" action="/students/add">
      <fieldset>
        <div class="row">

          <div class="col-md-6">
            <div class="form-group">
              <label>Enrolled in Course:</label>
              <% if (courses.length > 0) { %>
                <select class="form-control" name="course" id="course" required>
                  <option value="">Select Course</option>
                  <% courses.forEach(course => { %>
                    <option value="<%= course.courseId %>"><%= course.courseCode %></option>
                  <% }) %>
                </select>
              <% } else { %>
                <div>No Courses</div>
              <% } %>
            </div>
          </div>
        </div>
      </fieldset>

      <hr />

      <input
        type="submit"
        class="btn btn-primary float-right"
        value="Add Student"
      />
      <br /><br /><br />
    </form>
  </main>
</div>

```

WEB700_Assignment6_Asintha Dilshan Jayasekara (170388235)

Server.js

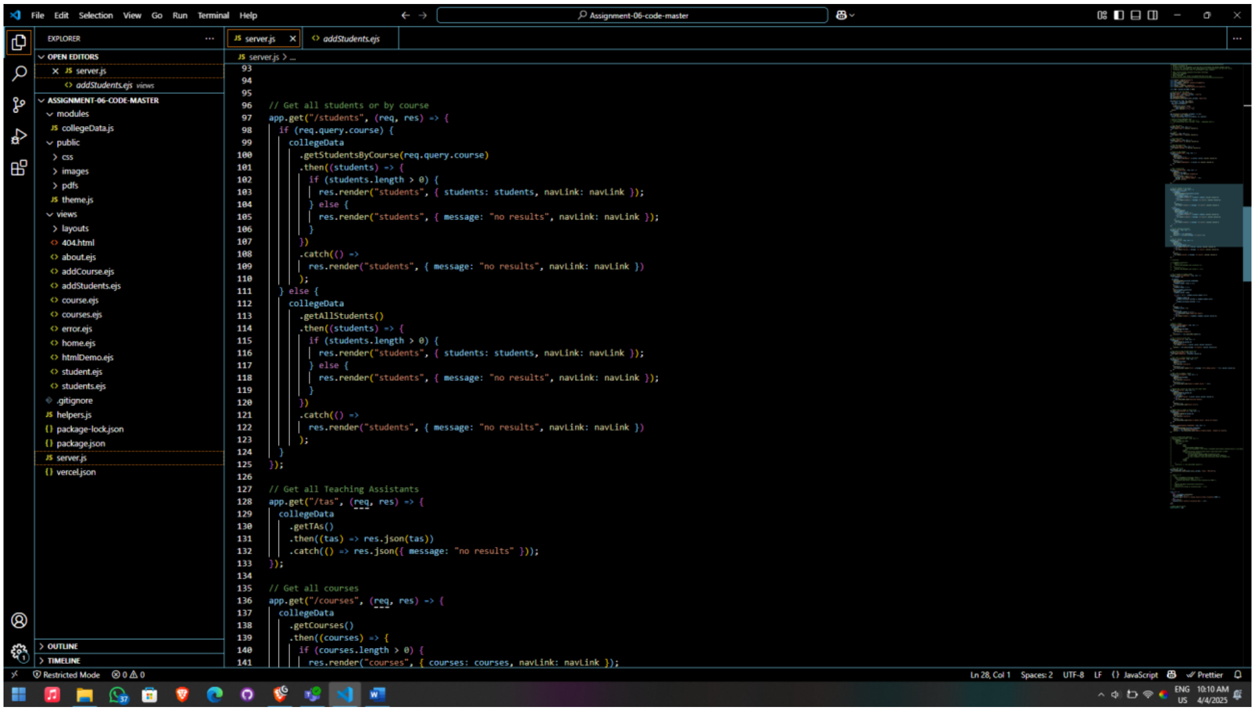


```javascript
/*********************************************************************************
*  WEB700 □ Assignment 06
*  I declare that this assignment is my own work in accordance with Seneca Academic Policy.
*  No part of this assignment has been copied manually or electronically from any other source
*  (including 3rd party web sites) or distributed to other students.
*
*  Name: Asintha Dilshan Jayasekara Wisurumana Arachchige
*  Student ID: 170388235
*  Date: 03/25/2025
*  Online (Vercel) Link: https://assignment-06-code.vercel.app/
********************************************************************************/

const express = require("express");
const path = require("path");
const collegeData = require("./modules/collegeData");
const app = express();
const helpers = require("./helpers");
const expressLayouts = require("express-ejs-layouts");

const PORT = process.env.PORT || 8080;

// Set EJS as the view engine
app.set("layout", "layouts/main");
app.set("views", path.join(__dirname, "views"));
app.set("view engine", "ejs");
app.use(expressLayouts);
app.use(express.static(path.join(__dirname, "public")));

app.use(function (req, res, next) {
  let route = req.path.substring(1);
  app.locals.activeRoute =
    "/" +
    (isNaN(route.split("/")[1])
      ? route.replace(/\/(?!.*)/, "")
      : route.replace(/\/(.*)/, ""));
  next();
});

app.use(express.urlencoded({ extended: true }));
const navLink = (url, options) =>
  helpers.navLink(app.locals.activeRoute, url, options);

// Route to Serve addStudent.html
// app.get("/students/add", (req, res) => {
//   res.sendFile(path.join(__dirname, "views", "addStudent.html"));
// });

// Home Page Route
app.get("/home", (req, res) => {
```

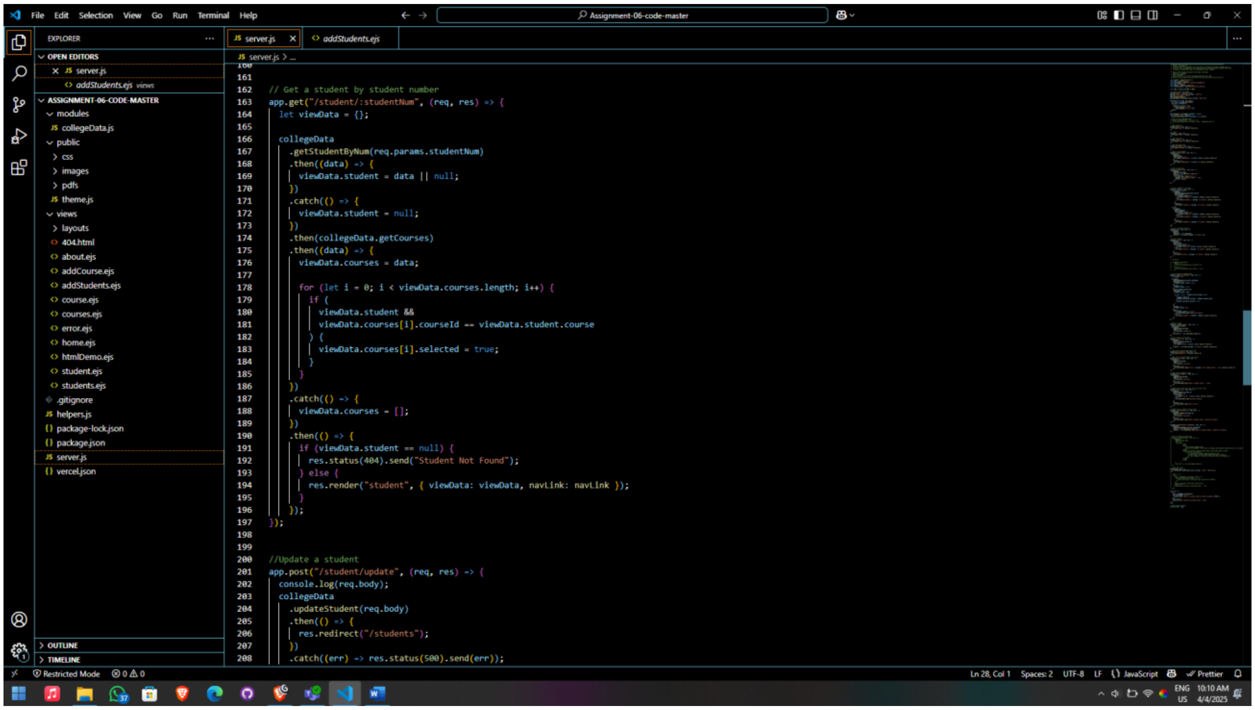


```javascript
// });

// Home Page Route
app.get("/home", (req, res) => {
  res.render("home", { navLink: navLink });
});

// / Route
app.get("/", (req, res) => {
  res.render("home", { navLink: navLink });
});

// About Page Route
app.get("/about", (req, res) => {
  res.render("about", { navLink: navLink });
});

// HTML Demo Page Route
app.get("/htmlDemo", (req, res) => {
  res.render("htmlDemo", { navLink: navLink });
});

// Route for Add Student
app.get("/students/add", (req, res) => {
  collegeData
    .getCourses()
    .then((courses) => {
      res.render("addStudents", { courses: courses, navLink: navLink });
    })
    .catch(() => {
      res.render("addStudents", { courses: [], navLink: navLink });
    });
});


// POST /students/add
app.post("/students/add", (req, res) => {
  collegeData
    .addStudent(req.body)
    .then(() => res.redirect("/students"))
    .catch((err) =>
      res.status(500).render("error", {
        message: "Error adding student: " + err,
        navLink: navLink,
      })
    );
});


```

WEB700_Assignment6_Asintha Dilshan Jayasekara (170388235)

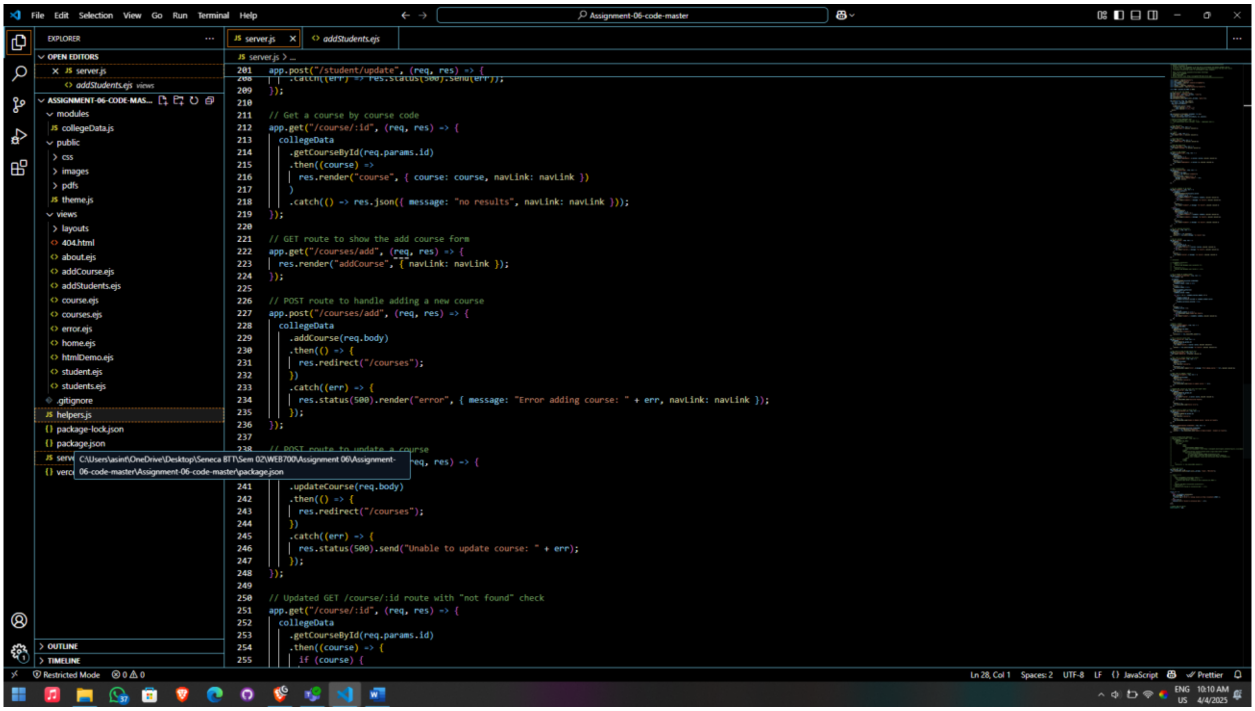


```javascript



// Get all students or by course
app.get("/students", (req, res) => {
  if (req.query.course) {
    collegeData
      .getStudentsByCourse(req.query.course)
      .then((students) => {
        if (students.length > 0) {
          res.render("students", { students: students, navLink: navLink });
        } else {
          res.render("students", { message: "no results", navLink: navLink });
        }
      })
      .catch(() =>
        res.render("students", { message: "no results", navLink: navLink })
      );
  } else {
    collegeData
      .getAllStudents()
      .then((students) => {
        if (students.length > 0) {
          res.render("students", { students: students, navLink: navLink });
        } else {
          res.render("students", { message: "no results", navLink: navLink });
        }
      })
      .catch(() =>
        res.render("students", { message: "no results", navLink: navLink })
      );
  }
});

// Get all Teaching Assistants
app.get("/tas", (req, res) => {
  collegeData
    .getTAs()
    .then((tas) => res.json(tas))
    .catch(() => res.json({ message: "no results" }));
});

// Get all courses
app.get("/courses", (req, res) => {
  collegeData
    .getCourses()
    .then((courses) => {
      if (courses.length > 0) {
        res.render("courses", { courses: courses, navLink: navLink });
```

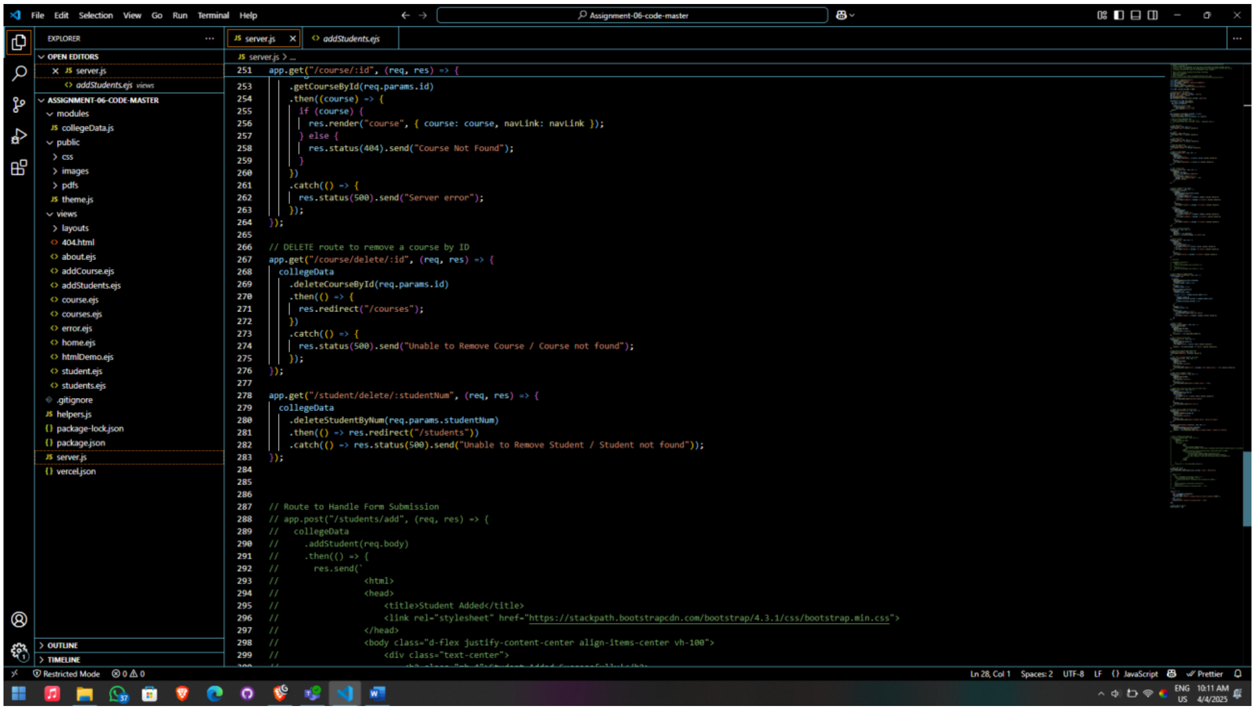


```javascript


// Get a student by student number
app.get("/student/:studentNum", (req, res) => {
  let viewData = {};

  collegeData
    .getStudentByNum(req.params.studentNum)
    .then((data) => {
      viewData.student = data || null;
    })
    .catch(() => {
      viewData.student = null;
    })
    .then(collegeData.getCourses)
    .then((data) => {
      viewData.courses = data;

      for (let i = 0; i < viewData.courses.length; i++) {
        if (
          viewData.student &&
          viewData.courses[i].courseId == viewData.student.course
        ) {
          viewData.courses[i].selected = true;
        }
      }
    })
    .catch(() => {
      viewData.courses = [];
    })
    .then(() => {
      if (viewData.student == null) {
        res.status(404).send("Student Not Found");
      } else {
        res.render("student", { viewData: viewData, navLink: navLink });
      }
    });
});


//Update a student
app.post("/student/update", (req, res) => {
  console.log(req.body);
  collegeData
    .updateStudent(req.body)
    .then(() => {
      res.redirect("/students");
    })
    .catch((err) => res.status(500).send(err));
```

WEB700_Assignment6_Asintha Dilshan Jayasekara (170388235)

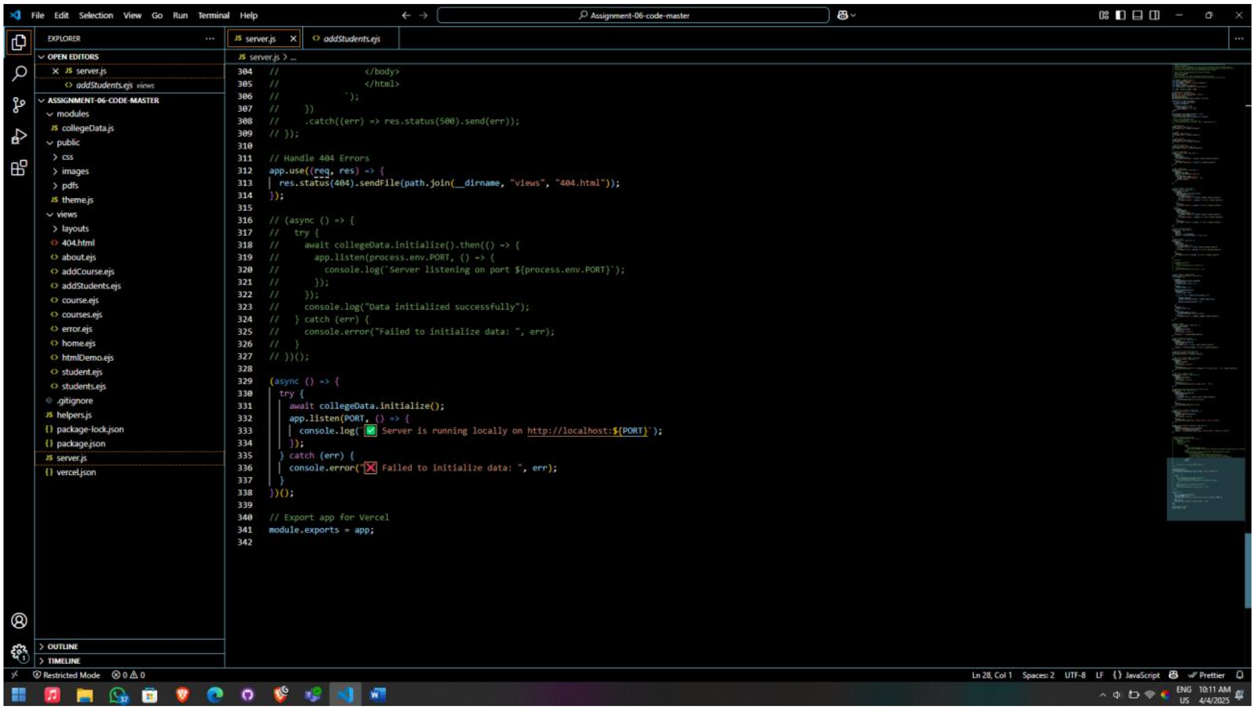


```
app.post("/student/update", (req, res) => {
    .catch((err) => res.status(500).send(err));
});

// Get a course by course code
app.get("/course/:id", (req, res) => {
  collegeData
    .getCourseById(req.params.id)
    .then((course) =>
      res.render("course", { course: course, navLink: navLink })
    )
    .catch(() => res.json({ message: "no results", navLink: navLink }));
});

// GET route to show the add course form
app.get("/courses/add", (req, res) => {
  res.render("addCourse", { navLink: navLink });
});

// POST route to handle adding a new course
app.post("/courses/add", (req, res) => {
  collegeData
    .addCourse(req.body)
    .then(() => {
      res.redirect("/courses");
    })
    .catch((err) => {
      res.status(500).render("error", { message: "Error adding course: " + err, navLink: navLink });
    });
});

// POST route to update a course
                          req, res) => {

    .updateCourse(req.body)
    .then(() => {
      res.redirect("/courses");
    })
    .catch((err) => {
      res.status(500).send("Unable to update course: " + err);
    });
});

// Updated GET /course/:id route with "not found" check
app.get("/course/:id", (req, res) => {
  collegeData
    .getCourseById(req.params.id)
    .then((course) => {
      if (course) {
```



```
app.get("/course/:id", (req, res) => {
    .getCourseById(req.params.id)
    .then((course) => {
      if (course) {
        res.render("course", { course: course, navLink: navLink });
      } else {
        res.status(404).send("Course Not Found");
      }
    })
    .catch(() => {
      res.status(500).send("Server error");
    });
});

// DELETE route to remove a course by ID
app.get("/course/delete/:id", (req, res) => {
  collegeData
    .deleteCourseById(req.params.id)
    .then(() => {
      res.redirect("/courses");
    })
    .catch(() => {
      res.status(500).send("Unable to Remove Course / Course not found");
    });
});

app.get("/student/delete/:studentNum", (req, res) => {
  collegeData
    .deleteStudentByNum(req.params.studentNum)
    .then(() => res.redirect("/students"))
    .catch(() => res.status(500).send("Unable to Remove Student / Student not found"));
});



// Route to Handle Form Submission
// app.post("/students/add", (req, res) => {
//   collegeData
//     .addStudent(req.body)
//     .then(() => {
//       res.send(`
//                 <html>
//                 <head>
//                     <title>Student Added</title>
//                     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
//                 </head>
//                 <body class="d-flex justify-content-center align-items-center vh-100">
//                     <div class="text-center">
```

WEB700_Assignment6_Asintha Dilshan Jayasekara (170388235)



```
//                </body>
//                </html>
//              `);
//      })
//      .catch((err) => res.status(500).send(err));
// });

// Handle 404 Errors
app.use((req, res) => {
  res.status(404).sendFile(path.join(__dirname, "views", "404.html"));
});

// (async () => {
//   try {
//     await collegeData.initialize().then(() => {
//       app.listen(process.env.PORT, () => {
//         console.log(`Server listening on port ${process.env.PORT}`);
//       });
//     });
//     console.log("Data initialized successfully");
//   } catch (err) {
//     console.error("Failed to initialize data: ", err);
//   }
// })();

(async () => {
  try {
    await collegeData.initialize();
    app.listen(PORT, () => {
      console.log(`✅ Server is running locally on http://localhost:${PORT}`);
    });
  } catch (err) {
    console.error("❌ Failed to initialize data: ", err);
  }
})();

// Export app for Vercel
module.exports = app;
```