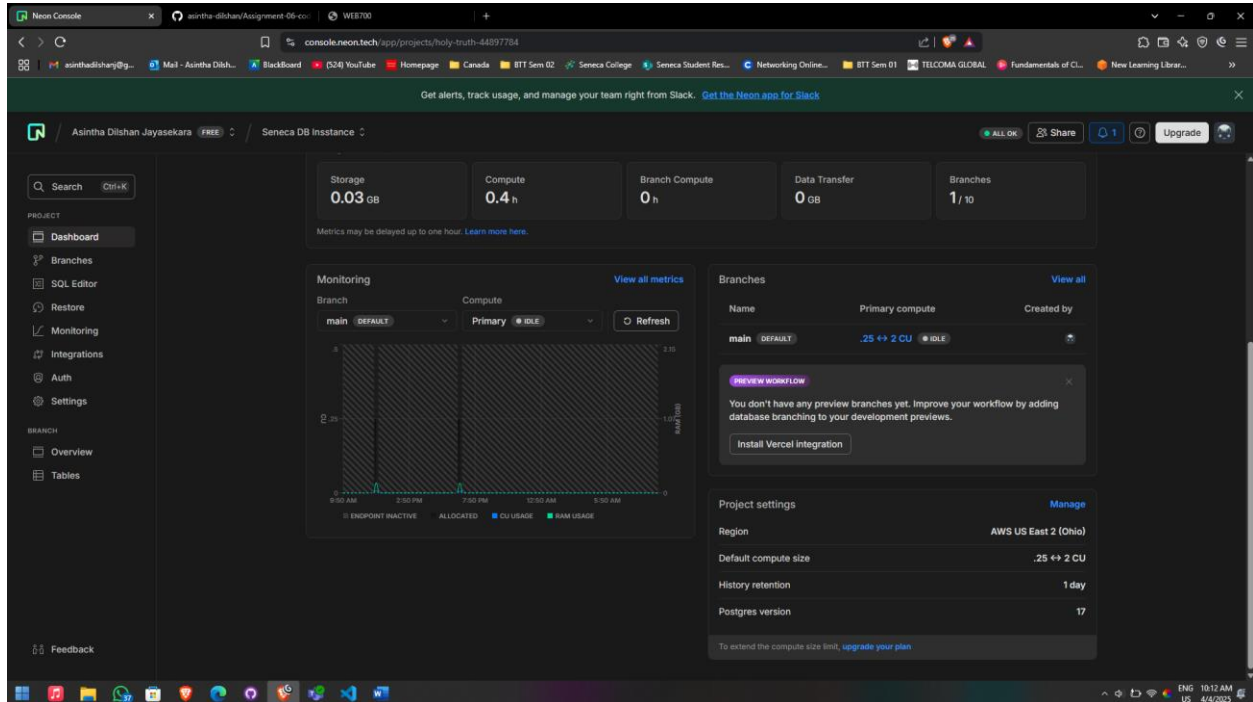


WEB700_Assignment6_Asintha Dilshan Jayasekara (170388235)

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

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

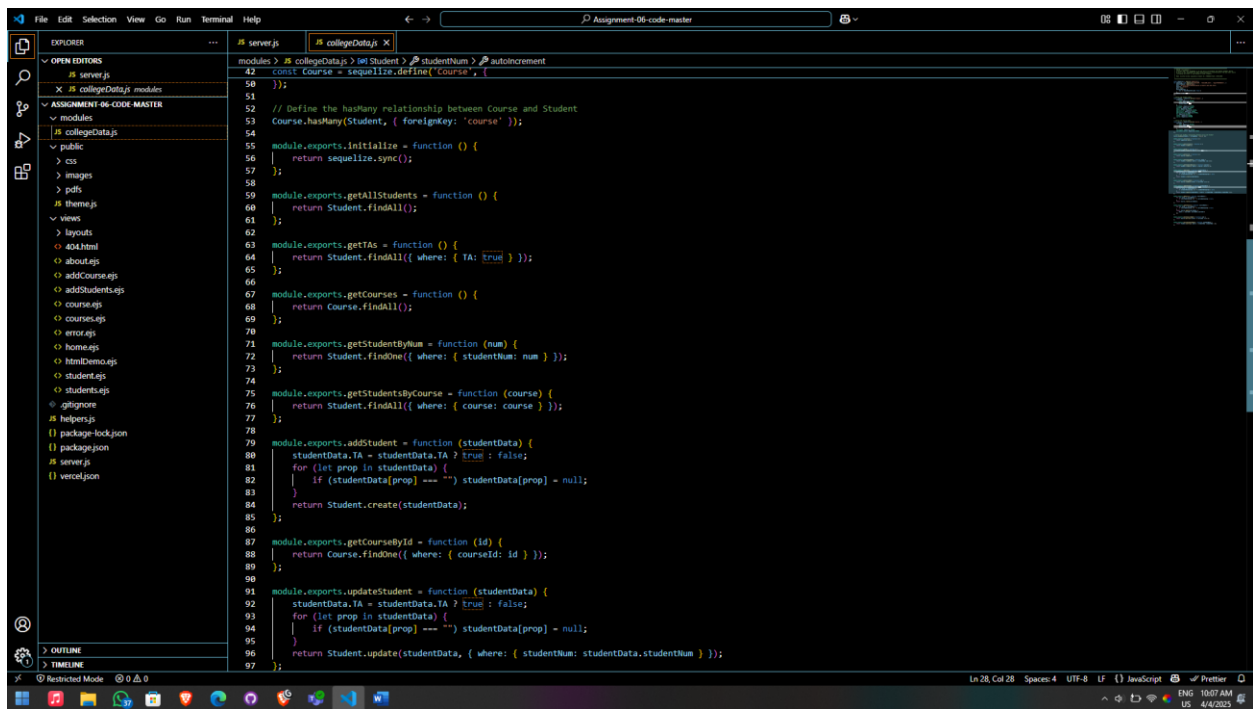
Neon.tech Databse



collegeData.js

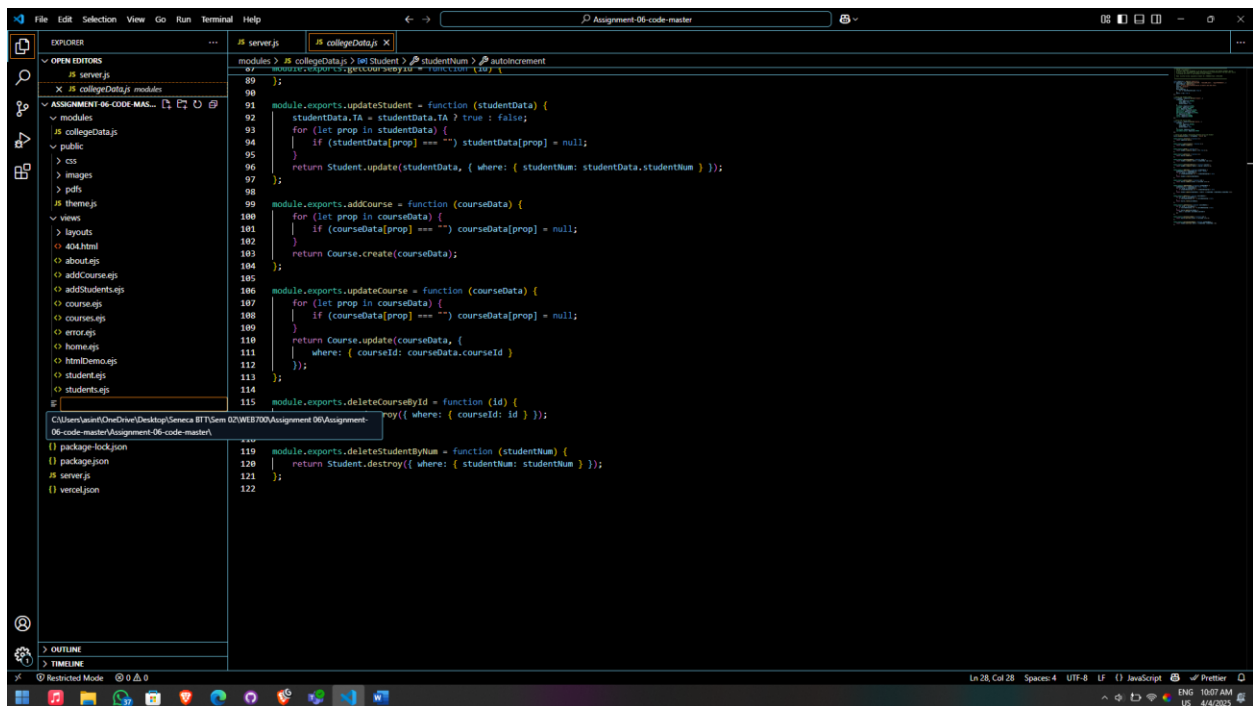
```
modules > collegeData.js > on Student > studentNum > autoincrement
1
2 * WEB700 Assignment 2
3 * I declare that this assignment is my own work in accordance with Seneca Academic Policy.
4 * No part of this assignment has been copied manually or electronically from any other source
5 * (including web sites) or distributed to other students.
6
7 * Name: Asintha Dilshan Jayasekara Student ID: 170388235 Date: 16/02/2025
8
9
10
11 const Sequelize = require('sequelize');
12 var sequelize = new Sequelize('SenecaDB', 'SenecaDB_owner', 'npg_8yn3KEULHx', {
13   dialectModule: require('pg'),
14   host: 'ep-still-band-aidsvdj-pooler-us-east-2.aws.neon.tech',
15   dialect: 'postgres',
16   port: 5432,
17   dialectOptions: {
18     ssl: { rejectUnauthorized: false }
19   },
20   query: { raw: true }
21 });
22
23 // Define the Student model
24 const Student = sequelize.define('Student', {
25   studentNum: {
26     type: Sequelize.INTEGER,
27     primaryKey: true,
28     autoIncrement: true
29   },
30   firstName: Sequelize.STRING,
31   lastName: Sequelize.STRING,
32   email: Sequelize.STRING,
33   addressStreet: Sequelize.STRING,
34   addressCity: Sequelize.STRING,
35   addressProvince: Sequelize.STRING,
36   TA: Sequelize.BOOLEAN,
37   status: Sequelize.STRING,
38   course: Sequelize.INTEGER
39 });
40
41 // Define the Course model
42 const Course = sequelize.define('Course', {
43   courseId: {
44     type: Sequelize.INTEGER,
45     primaryKey: true,
46     autoIncrement: true
47   },
48   courseCode: Sequelize.STRING,
49   courseDescription: Sequelize.STRING
```

WEB700_Assignment6_Asintha Dilshan Jayasekara (170388235)



This screenshot shows the first part of the `server.js` file in a VS Code editor. The Explorer sidebar on the left shows the project structure, including `server.js`, `collegeData.js`, and various static files. The main editor displays the following code:

```
modules > # collegeData.js > set Student > # studentNum > # autoincrement
42 const Course = sequelize.define('course', {
43   title: { type: String, required: true },
44   credits: { type: Integer, required: true },
45   prereqs: { type: String, required: false }
46 });
47
48 // Define the hasMany relationship between Course and Student
49 Course.hasMany(Student, { foreignKey: 'course' });
50
51 module.exports.initialize = function () {
52   return sequelize.sync();
53 };
54
55 module.exports.getAllStudents = function () {
56   return Student.findAll();
57 };
58
59 module.exports.getTAs = function () {
60   return Student.findAll({ where: { TA: true } });
61 };
62
63 module.exports.getCourses = function () {
64   return Course.findAll();
65 };
66
67 module.exports.getStudentsByCourse = function (course) {
68   return Student.findAll({ where: { course: course } });
69 };
70
71 module.exports.addStudent = function (studentData) {
72   studentData.TA = studentData.TA ? true : false;
73   for (let prop in studentData) {
74     if (studentData[prop] === '') studentData[prop] = null;
75   }
76   return Student.create(studentData);
77 };
78
79 module.exports.getStudentById = function (id) {
80   return Student.findOne({ where: { id: id } });
81 };
82
83 module.exports.updateStudent = function (studentData) {
84   studentData.TA = studentData.TA ? true : false;
85   for (let prop in studentData) {
86     if (studentData[prop] === '') studentData[prop] = null;
87   }
88   return Student.update(studentData, { where: { studentNum: studentData.studentNum } });
89 };
90
```



This screenshot shows the second part of the `server.js` file in a VS Code editor. The Explorer sidebar on the left shows the project structure, including `server.js`, `collegeData.js`, and various static files. The main editor displays the following code:

```
modules > # collegeData.js > set Student > # studentNum > # autoincrement
89
90 module.exports.updateCourse = function (courseData) {
91   for (let prop in courseData) {
92     if (courseData[prop] === '') courseData[prop] = null;
93   }
94   return Course.update(courseData, { where: { courseId: courseData.courseId } });
95 };
96
97 module.exports.deleteCourseById = function (id) {
98   return Course.destroy({ where: { id: id } });
99 };
100
101 module.exports.deleteStudentById = function (studentNum) {
102   return Student.destroy({ where: { studentNum: studentNum } });
103 };
104
105 module.exports.deleteCourseByTitle = function (title) {
106   return Course.destroy({ where: { title: title } });
107 };
108
109 module.exports.deleteStudentByCourse = function (course) {
110   return Student.destroy({ where: { course: course } });
111 };
112
113 module.exports.deleteCourseByCredits = function (credits) {
114   return Course.destroy({ where: { credits: credits } });
115 };
116
117 module.exports.deleteCourseByPrereqs = function (prereqs) {
118   return Course.destroy({ where: { prereqs: prereqs } });
119 };
120
121 module.exports.deleteCourseByAll = function () {
122   return Course.destroy({ where: {} });
123 };
124
```

addCourse.ejs

```

1 <div class="row">
2   <main class="col-md-12">
3     <hr />
4     <h2>Add a New Course</h2>
5     <hr />
6
7     <form method="post" action="/courses/add">
8       <fieldset>
9         <legend>Course Information</legend>
10        <div class="row">
11          <div class="col-md-6">
12            <div class="form-group">
13              <label>Course Code:</label>
14              <input class="form-control" name="courseCode" type="text" required />
15            </div>
16          </div>
17          <div class="col-md-6">
18            <div class="form-group">
19              <label>Course Description:</label>
20              <input class="form-control" name="courseDescription" type="text" required />
21            </div>
22          </div>
23        </div>
24      </fieldset>
25
26      <hr />
27      <input type="submit" class="btn btn-primary float-right" value="Add Course" />
28    </form>
29  </main>
30 </div>
31
32

```

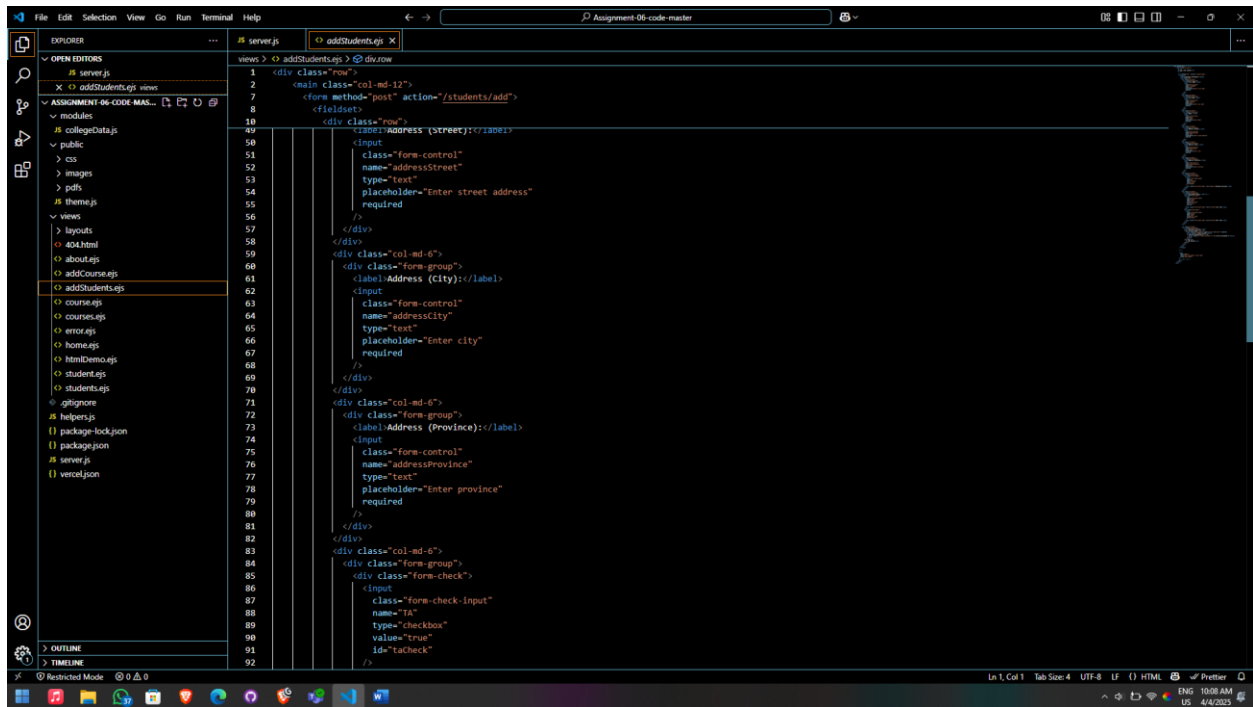
addStudent.ejs

```

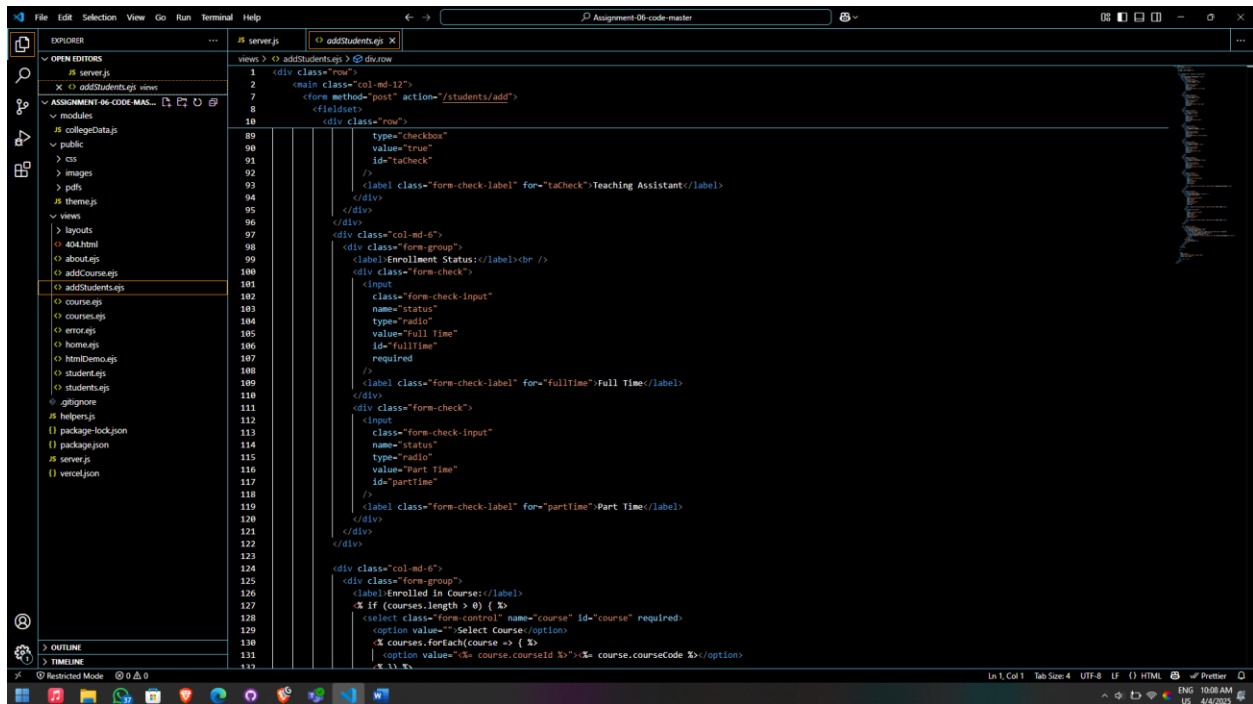
1 <div class="row">
2   <main class="col-md-12">
3     <hr />
4     <h2>Add a New Student</h2>
5     <hr />
6
7     <form method="post" action="/students/add">
8       <fieldset>
9         <legend>Personal Information</legend>
10        <div class="row">
11          <div class="col-md-6">
12            <div class="form-group">
13              <label>First Name:</label>
14              <input
15                class="form-control"
16                name="firstName"
17                type="text"
18                placeholder="Enter first name"
19                required
20              />
21            </div>
22          </div>
23          <div class="col-md-6">
24            <div class="form-group">
25              <label>Last Name:</label>
26              <input
27                class="form-control"
28                name="lastName"
29                type="text"
30                placeholder="Enter last name"
31                required
32              />
33            </div>
34          </div>
35        </div>
36        <div class="col-md-6">
37          <div class="form-group">
38            <label>Email:</label>
39            <input
40              class="form-control"
41              name="email"
42              type="email"
43              placeholder="Enter email"
44              required
45            />
46          </div>
47        </div>
48        <div class="col-md-6">
49          <div class="form-group">
50            <label>Address (Street):</label>

```

WEB700_Assignment6_Asintha Dilshan Jayasekara (170388235)

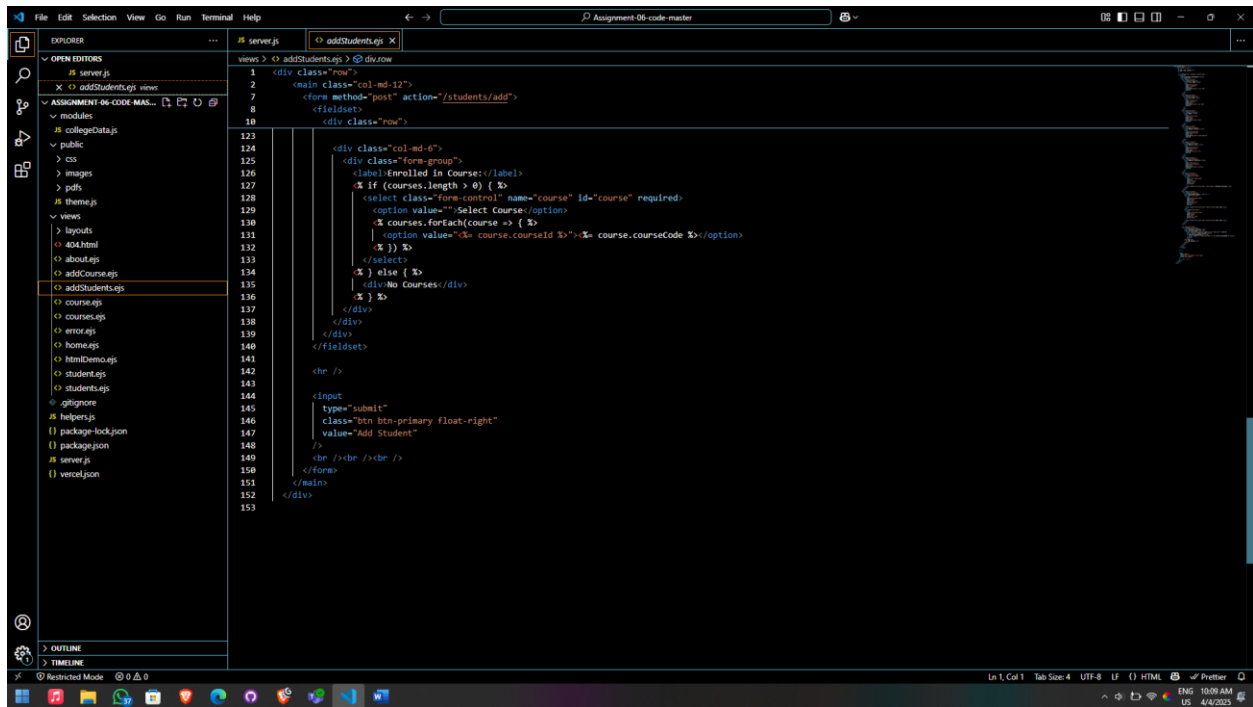


```
1 <div class="row">
2   <main class="col-md-12">
3     <form method="post" action="/students/add">
4       <fieldset>
5         <div class="row">
6           <div class="col-md-12">
7             <div class="form-group">
8               <label>Address (Street):</label>
9               <input
10                class="form-control"
11                name="addressStreet"
12                type="text"
13                placeholder="Enter street address"
14                required
15              />
16            </div>
17            <div class="col-md-6">
18              <div class="form-group">
19                <label>Address (City):</label>
20                <input
21                  class="form-control"
22                  name="addressCity"
23                  type="text"
24                  placeholder="Enter city"
25                  required
26                />
27              </div>
28            </div>
29            <div class="col-md-6">
30              <div class="form-group">
31                <label>Address (Province):</label>
32                <input
33                  class="form-control"
34                  name="addressProvince"
35                  type="text"
36                  placeholder="Enter province"
37                  required
38                />
39              </div>
40            </div>
41          </div>
42          <div class="col-md-6">
43            <div class="form-group">
44              <div class="form-check">
45                <input
46                  class="form-check-input"
47                  name="TA"
48                  type="checkbox"
49                  value="true"
50                  id="taCheck"
51                />
52                <label class="form-check-label" for="taCheck">Teaching Assistant</label>
53              </div>
54            </div>
55          </div>
56        </div>
57        <div class="col-md-6">
58          <div class="form-group">
59            <label>Enrollment Status:</label><br />
60            <div class="form-check">
61              <input
62                class="form-check-input"
63                name="status"
64                type="radio"
65                value="Full Time"
66                id="fullTime"
67                required
68              />
69              <label class="form-check-label" for="fullTime">Full Time</label>
70            </div>
71            <div class="form-check">
72              <input
73                class="form-check-input"
74                name="status"
75                type="radio"
76                value="Part Time"
77                id="partTime"
78              />
79              <label class="form-check-label" for="partTime">Part Time</label>
80            </div>
81          </div>
82        </div>
83      </div>
84      <div class="col-md-6">
85        <div class="form-group">
86          <label>Enrolled in Course:</label>
87          <div class="form-check">
88            <input
89              class="form-check-input"
90              name="course"
91              type="radio"
92              value="Full Time"
93              id="fullTime"
94              required
95            />
96            <label class="form-check-label" for="fullTime">Full Time</label>
97          </div>
98          <div class="form-check">
99            <input
100              class="form-check-input"
101              name="status"
102              type="radio"
103              value="Part Time"
104              id="partTime"
105            />
106            <label class="form-check-label" for="partTime">Part Time</label>
107          </div>
108        </div>
109      </div>
110    </form>
111  </main>
112 </div>
113 </div>
```



```
1 <div class="row">
2   <main class="col-md-12">
3     <form method="post" action="/students/add">
4       <fieldset>
5         <div class="row">
6           <div class="col-md-12">
7             <div class="form-group">
8               <label>Address (Street):</label>
9               <input
10                class="form-control"
11                name="addressStreet"
12                type="text"
13                placeholder="Enter street address"
14                required
15              />
16            </div>
17            <div class="col-md-6">
18              <div class="form-group">
19                <label>Address (City):</label>
20                <input
21                  class="form-control"
22                  name="addressCity"
23                  type="text"
24                  placeholder="Enter city"
25                  required
26                />
27              </div>
28            </div>
29            <div class="col-md-6">
30              <div class="form-group">
31                <label>Address (Province):</label>
32                <input
33                  class="form-control"
34                  name="addressProvince"
35                  type="text"
36                  placeholder="Enter province"
37                  required
38                />
39              </div>
40            </div>
41          </div>
42          <div class="col-md-6">
43            <div class="form-group">
44              <div class="form-check">
45                <input
46                  class="form-check-input"
47                  name="TA"
48                  type="checkbox"
49                  value="true"
50                  id="taCheck"
51                />
52                <label class="form-check-label" for="taCheck">Teaching Assistant</label>
53              </div>
54            </div>
55          </div>
56        </div>
57        <div class="col-md-6">
58          <div class="form-group">
59            <label>Enrollment Status:</label><br />
60            <div class="form-check">
61              <input
62                class="form-check-input"
63                name="status"
64                type="radio"
65                value="Full Time"
66                id="fullTime"
67                required
68              />
69              <label class="form-check-label" for="fullTime">Full Time</label>
70            </div>
71            <div class="form-check">
72              <input
73                class="form-check-input"
74                name="status"
75                type="radio"
76                value="Part Time"
77                id="partTime"
78              />
79              <label class="form-check-label" for="partTime">Part Time</label>
80            </div>
81          </div>
82        </div>
83      </div>
84      <div class="col-md-6">
85        <div class="form-group">
86          <label>Enrolled in Course:</label>
87          <div class="form-check">
88            <input
89              class="form-check-input"
90              name="course"
91              type="radio"
92              value="Full Time"
93              id="fullTime"
94              required
95            />
96            <label class="form-check-label" for="fullTime">Full Time</label>
97          </div>
98          <div class="form-check">
99            <input
100              class="form-check-input"
101              name="status"
102              type="radio"
103              value="Part Time"
104              id="partTime"
105            />
106            <label class="form-check-label" for="partTime">Part Time</label>
107          </div>
108        </div>
109      </div>
110    </form>
111  </main>
112 </div>
113 </div>
```

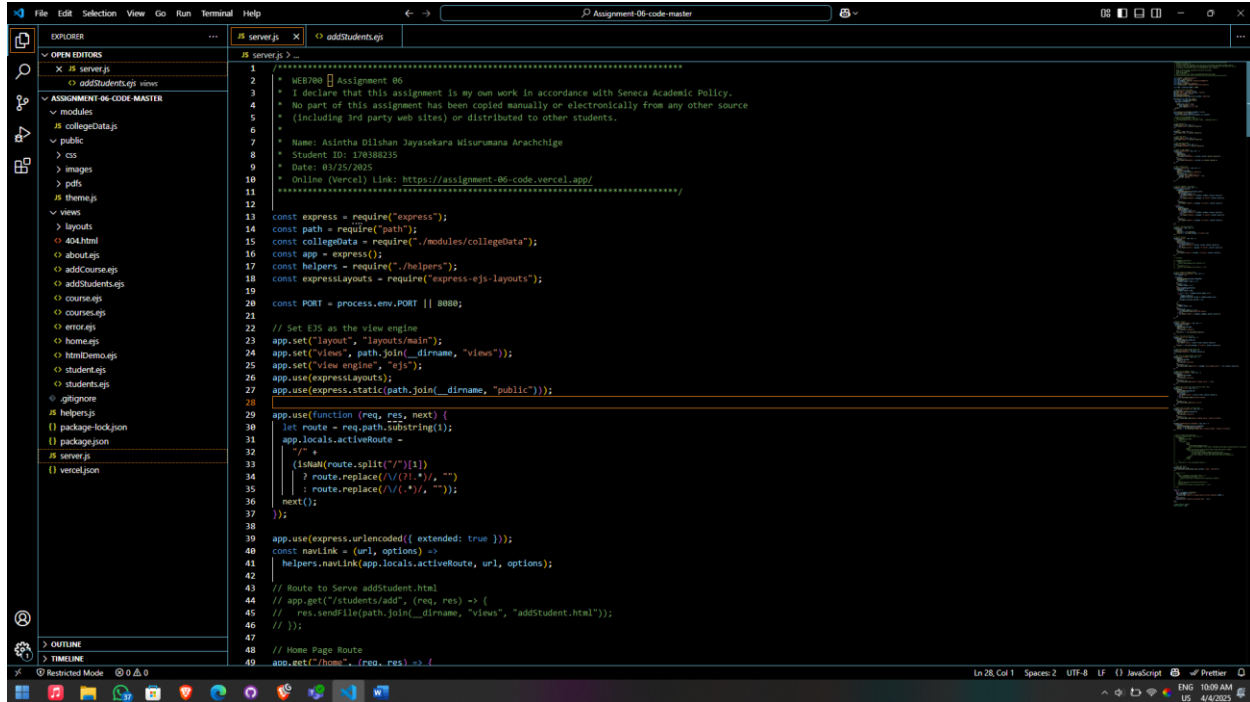
WEB700_Assignment6_Asintha Dilshan Jayasekara (170388235)



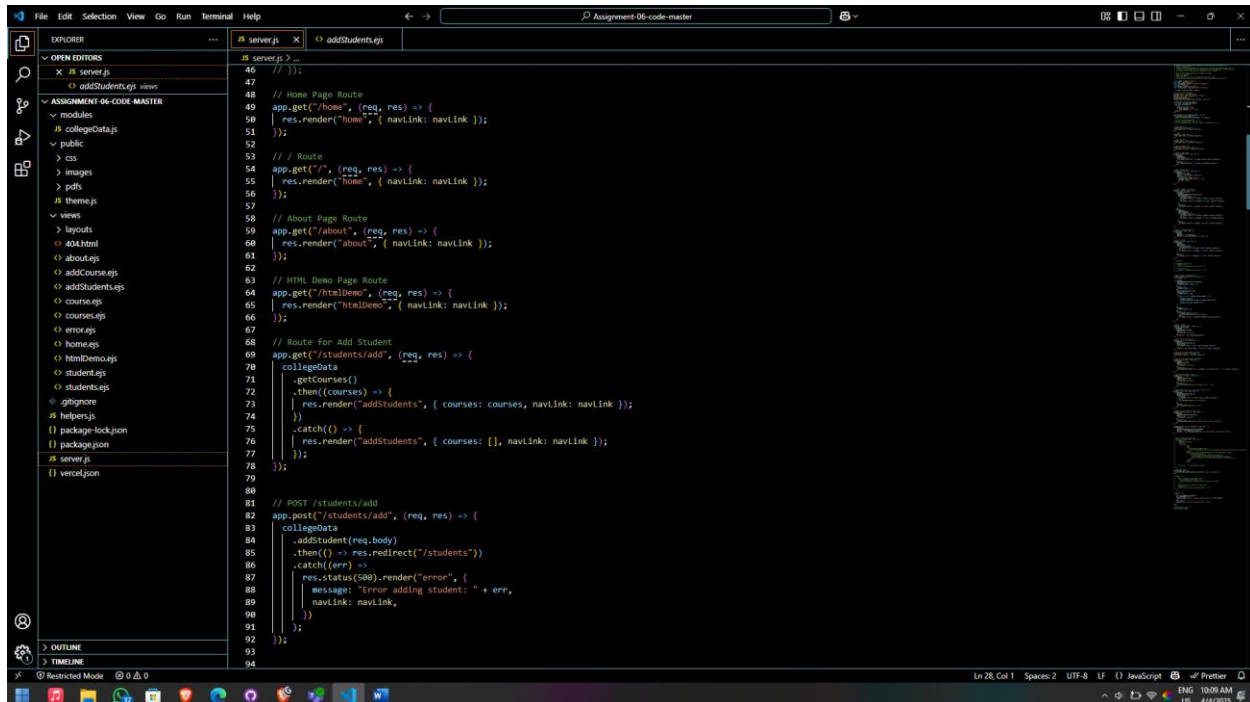
```
1 <div class="row">
2   <main class="col-md-12">
3     <form method="post" action="/students/add">
4       <fieldset>
5         <div class="row">
6           <div class="col-md-6">
7             <div class="form-group">
8               <label>Enrolled in Course:</label>
9               <div if (courses.length > 0) { %>
10                 <select class="form-control" name="course" id="course" required>
11                   <option value="">Select Course</option>
12                   <% courses.forEach(course => { %>
13                     <option value="<%= course.courseId %%"><%= course.courseCode %%</option>
14                   <% } %>
15                 </select>
16               <%> else { %>
17                 <div>No Courses</div>
18               <%> %>
19             </div>
20           </div>
21         </div>
22       </fieldset>
23     </form>
24   </main>
25 </div>
```

WEB700_Assignment6_Asintha Dilshan Jayasekara (170388235)

Server.js



```
1 //=====
2 * WEB700 Assignment 06
3 * I declare that this assignment is my own work in accordance with Seneca Academic Policy.
4 * No part of this assignment has been copied manually or electronically from any other source
5 * (including 3rd party web sites) or distributed to other students.
6 *
7 * Name: Asintha Dilshan Jayasekara Misurumana Arachchige
8 * Student ID: 170388235
9 * Date: 01/25/2025
10 * Online (Vercel) Link: https://assignment-06-code.vercel.app/
11 //=====
12
13 const express = require("express");
14 const path = require("path");
15 const collegeData = require("../modules/collegeData");
16 const app = express();
17 const helpers = require("../helpers");
18 const expressLayouts = require("express-ejs-layouts");
19
20 const PORT = process.env.PORT || 8080;
21
22 // Set EJS as the view engine
23 app.set("layout", "layouts/main");
24 app.set("views", path.join(__dirname, "views"));
25 app.set("view engine", "ejs");
26 app.use(express.static(path.join(__dirname, "public")));
27
28
29 app.use(function (req, res, next) {
30   let route = req.path.substring(1);
31   app.locals.activeRoute =
32     "/" +
33     (isNaN(route.split("/") [1])
34       ? route.replace(/\/?/, "")
35       : route.replace(/\/?/, "/"));
36   next();
37 });
38
39 app.use(express.urlencoded({ extended: true }));
40 const navlink = {url, options =>
41   helpers.navlink(app.locals.activeRoute, url, options);
42 }
43
44 // Route to Serve addStudent.html
45 // app.get("/students/add", (req, res) => {
46 //   res.sendFile(path.join(__dirname, "views", "addStudent.html"));
47 // });
48
49 // Home Page Route
```



```
46 // });
47
48 // Home Page Route
49 app.get("/home", (req, res) => {
50   res.render("home", { navlink: navlink });
51 });
52
53 // / Route
54 app.get("/", (req, res) => {
55   res.render("home", { navlink: navlink });
56 });
57
58 // About Page Route
59 app.get("/about", (req, res) => {
60   res.render("about", { navlink: navlink });
61 });
62
63 // HTML Demo Page Route
64 app.get("/htmlDemo", (req, res) => {
65   res.render("htmlDemo", { navlink: navlink });
66 });
67
68 // Route for Add Student
69 app.get("/students/add", (req, res) => {
70   collegeData
71     .getCourses()
72     .then((courses) => {
73       res.render("addStudents", { courses: courses, navlink: navlink });
74     })
75     .catch(() => {
76       res.render("addStudents", { courses: [], navlink: navlink });
77     });
78 });
79
80
81 // POST /students/add
82 app.post("/students/add", (req, res) => {
83   collegeData
84     .addStudent(req.body)
85     .then(() => res.redirect("/students"))
86     .catch(err) =>
87       res.status(500).render("error", {
88         message: "Error adding student: " + err,
89         navlink: navlink,
90       });
91 });
92
93
94
```

WEB700_Assignment6_Asintha Dilshan Jayasekara (170388235)

```
13 // server.js
14
15 // Get all students or by course
16 app.get("/students", (req, res) => {
17   if (req.query.course) {
18     collegeData
19       .getStudentsByCourse(req.query.course)
20       .then((students) => {
21         if (students.length > 0) {
22           res.render("students", { students: students, navlink: navlink });
23         } else {
24           res.render("students", { message: "no results", navlink: navlink });
25         }
26       })
27       .catch(() => {
28         res.render("students", { message: "no results", navlink: navlink });
29       });
30   } else {
31     collegeData
32       .getAllStudents()
33       .then((students) => {
34         if (students.length > 0) {
35           res.render("students", { students: students, navlink: navlink });
36         } else {
37           res.render("students", { message: "no results", navlink: navlink });
38         }
39       })
40       .catch(() => {
41         res.render("students", { message: "no results", navlink: navlink });
42       });
43   }
44 });
45
46 // Get all Teaching Assistants
47 app.get("/tas", (req, res) => {
48   collegeData
49     .getTas()
50     .then((tas) => res.json(tas))
51     .catch(() => res.json({ message: "no results" }));
52 });
53
54 // Get all courses
55 app.get("/courses", (req, res) => {
56   collegeData
57     .getCourses()
58     .then((courses) => {
59       if (courses.length > 0) {
60         res.render("courses", { courses: courses, navlink: navlink });
61       }
62     })
63     .catch(() => {
64       res.render("courses", { message: "no results", navlink: navlink });
65     });
66 });
```

```
161 // server.js
162
163 // Get a student by student number
164 app.get("/student/:studentNum", (req, res) => {
165   let viewData = {};
166   collegeData
167     .getStudentByNum(req.params.studentNum)
168     .then((data) => {
169       viewData.student = data || null;
170     })
171     .catch(() => {
172       viewData.student = null;
173     })
174     .then(() => {
175       collegeData.getCourses()
176       .then((data) => {
177         viewData.courses = data;
178         for (let i = 0; i < viewData.courses.length; i++) {
179           if (
180             viewData.student &&
181             viewData.courses[i].courseId == viewData.student.course
182           ) {
183             viewData.courses[i].selected = true;
184           }
185         }
186       })
187       .catch(() => {
188         viewData.courses = [];
189       });
190     })
191     .then(() => {
192       if (viewData.student == null) {
193         res.status(404).send("Student Not Found");
194       } else {
195         res.render("student", { viewData: viewData, navlink: navlink });
196       }
197     });
198 });
199
200 // Update a student
201 app.post("/student/update", (req, res) => {
202   console.log(req.body);
203   collegeData
204     .updateStudent(req.body)
205     .then(() => {
206       res.redirect("/students");
207     })
208     .catch((err) => res.status(500).send(err));
209 });
```

WEB700_Assignment6_Asintha Dilshan Jayasekara (170388235)

```
server.js
addStudents.js
modules
collegeData.js
public
css
images
pdfs
theme.js
views
layouts
404.html
about.js
addCourse.js
addStudents.js
courses.js
courses.js
error.js
home.js
htmlDemo.js
student.js
students.js
.githignore
help.js
package-lock.json
package.json
server.js
vercel.json
```

```
server.js
201 app.post("/student/update", (req, res) => {
202   // Catch the error
203   .catch(err => res.status(500).send(err));
204 });
205
206 // Get a course by course code
207 app.get("/course/:id", (req, res) => {
208   collegeData
209   .getCourseById(req.params.id)
210   .then(course => {
211     res.render("course", { course: course, navLink: navLink });
212   })
213   .catch(() => res.json({ message: "no results", navLink: navLink }));
214 });
215
216 // GET route to show the add course form
217 app.get("/courses/add", (req, res) => {
218   res.render("addCourse", { navLink: navLink });
219 });
220
221 // POST route to handle adding a new course
222 app.post("/courses/add", (req, res) => {
223   collegeData
224   .addCourse(req.body)
225   .then(() => {
226     res.redirect("/courses");
227   })
228   .catch(err => {
229     res.status(500).render("error", { message: "Error adding course: " + err, navLink: navLink });
230   });
231 });
232
233 // POST route to update a course
234 app.post("/course/:id", (req, res) => {
235   collegeData
236   .updateCourse(req.body)
237   .then(() => {
238     res.redirect("/courses");
239   })
240   .catch(err => {
241     res.status(500).send("Unable to update course: " + err);
242   });
243 });
244
245 // Updated GET /course/:id route with "not found" check
246 app.get("/course/:id", (req, res) => {
247   collegeData
248   .getCourseById(req.params.id)
249   .then(course => {
250     if (course) {
251       res.render("course", { course: course, navLink: navLink });
252     } else {
253       res.status(404).send("Course Not Found");
254     }
255   })
256   .catch(() => {
257     res.status(500).send("Server error");
258   });
259 });
260
261 // DELETE route to remove a course by ID
262 app.delete("/course/delete/:id", (req, res) => {
263   collegeData
264   .deleteCourseById(req.params.id)
265   .then(() => {
266     res.redirect("/courses");
267   })
268   .catch(() => {
269     res.status(500).send("Unable to Remove Course / Course not found");
270   });
271 });
272
273 app.get("/student/delete/:studentId", (req, res) => {
274   collegeData
275   .deleteStudentById(req.params.studentId)
276   .then(() => res.redirect("/students"))
277   .catch(() => res.status(500).send("Unable to Remove Student / Student not found"));
278 });
279
280 // Route to Handle Form Submission
281 app.post("/students/add", (req, res) => {
282   collegeData
283   .addStudent(req.body)
284   .then(() => {
285     res.send()
286   })
287   .catch(err => {
288     res.status(500).send("Error adding student: " + err);
289   });
290 });
291
292 // HTML template for adding a student
293 // <html>
294 // <head>
295 // <title>Student Added</title>
296 // <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
297 // </head>
298 // <body class="d-flex justify-content-center align-items-center vh-100">
299 // <div class="text-center">
```

```
server.js
addStudents.js
modules
collegeData.js
public
css
images
pdfs
theme.js
views
layouts
404.html
about.js
addCourse.js
addStudents.js
courses.js
courses.js
error.js
home.js
htmlDemo.js
student.js
students.js
.githignore
help.js
package-lock.json
package.json
server.js
vercel.json
```

```
server.js
261 app.get("/course/:id", (req, res) => {
262   collegeData
263   .getCourseById(req.params.id)
264   .then(course => {
265     if (course) {
266       res.render("course", { course: course, navLink: navLink });
267     } else {
268       res.status(404).send("Course Not Found");
269     }
270   })
271   .catch(() => {
272     res.status(500).send("Server error");
273   });
274 });
275
276 // DELETE route to remove a course by ID
277 app.delete("/course/delete/:id", (req, res) => {
278   collegeData
279   .deleteCourseById(req.params.id)
280   .then(() => {
281     res.redirect("/courses");
282   })
283   .catch(() => {
284     res.status(500).send("Unable to Remove Course / Course not found");
285   });
286 });
287
288 app.get("/student/delete/:studentId", (req, res) => {
289   collegeData
290   .deleteStudentById(req.params.studentId)
291   .then(() => res.redirect("/students"))
292   .catch(() => res.status(500).send("Unable to Remove Student / Student not found"));
293 });
294
295 // Route to Handle Form Submission
296 app.post("/students/add", (req, res) => {
297   collegeData
298   .addStudent(req.body)
299   .then(() => {
300     res.send()
301   })
302   .catch(err => {
303     res.status(500).send("Error adding student: " + err);
304   });
305 });
306
307 // HTML template for adding a student
308 // <html>
309 // <head>
310 // <title>Student Added</title>
311 // <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
312 // </head>
313 // <body class="d-flex justify-content-center align-items-center vh-100">
314 // <div class="text-center">
```


WEB700_Assignment6_Asintha Dilshan Jayasekara (170388235)

```
server.js
304 // </body>
305 // </html>
306 // })
307 // })
308 // .catch(err) => res.status(500).send(err));
309 // });
310
311 // Handle 404 Errors
312 app.use((req, res) => {
313   res.status(404).sendFile(path.join(__dirname, "views", "404.html"));
314 });
315
316 // (async () => {
317 //   try {
318 //     await collegeData.initialize().then(() => {
319 //       app.listen(process.env.PORT, () => {
320 //         console.log("Server listening on port ${process.env.PORT}");
321 //       });
322 //     });
323 //     console.log("Data initialized successfully");
324 //   } catch (err) {
325 //     console.error("Failed to initialize data: ", err);
326 //   }
327 // })();
328
329 (async () => {
330   try {
331     await collegeData.initialize();
332     app.listen(PORT, () => {
333       console.log("Server is running locally on http://localhost:${PORT}");
334     });
335   } catch (err) {
336     console.error("Failed to initialize data: ", err);
337   }
338 })();
339
340 // Export app for Vercel
341 module.exports = app;
342
```