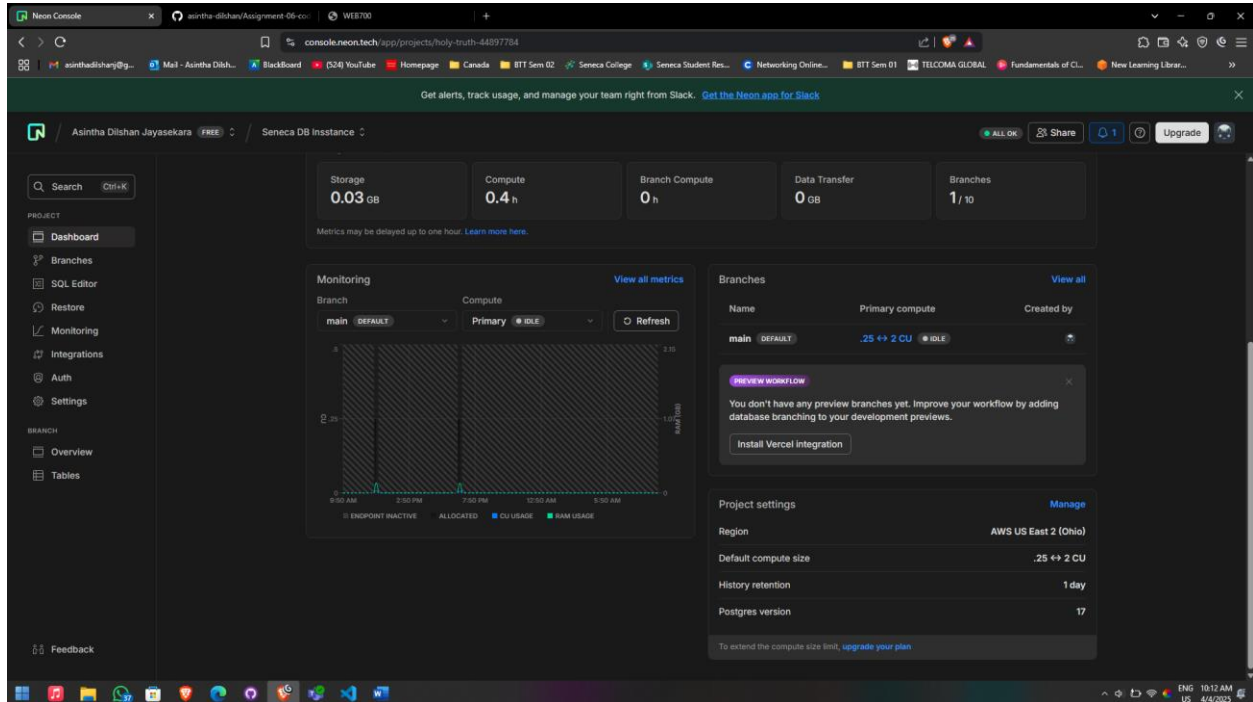


WEB700_Assignment6_Asintha Dilshan Jayasekara (170388235)

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

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

Neon.tech Databse



The screenshot shows the 'Tables' view in the Neon Console. The table 'Students' is selected, and its data is displayed in a table format. The table has the following columns: courseCode, courseDescription, createdAt, updatedAt, and Students.

courseCode	courseDescription	createdAt	updatedAt	Students
1	Test	2025-04-02 04:34:32.93...	2025-04-02 04:34:32.93...	Students
3	Web700	2025-04-04 14:02:07.72...	2025-04-04 14:02:07.72...	Students

WEB700_Assignment6_Asintha Dilshan Jayasekara (170388235)

Launch AWS Academy Learner Lab | MongoDB command not found | MongoDB Shell Download | MongoDB | MongoDB Shell Download | MongoDB | Neon Console

console.neon.tech/app/projects/holy-truth-44897784/branches/br-solitary-base-a50v3tqu/tables?database=SenecaDB

Get alerts, track usage, and manage your team right from Slack. [Get the Neon app for Slack](#)

Asintha Dilshan Jayasekara (FREE) | Seneca DB Instance | main

ALL OK | Share | 1 | Upgrade

Search | Ctrl+K

PROJECT

- Dashboard
- Branches
- SQL Editor
- Restore
- Monitoring
- Integrations
- Auth
- Settings

BRANCH

- Overview
- Tables

Feedback

Tables

SenecaDB

public

Search...

Courses

Students

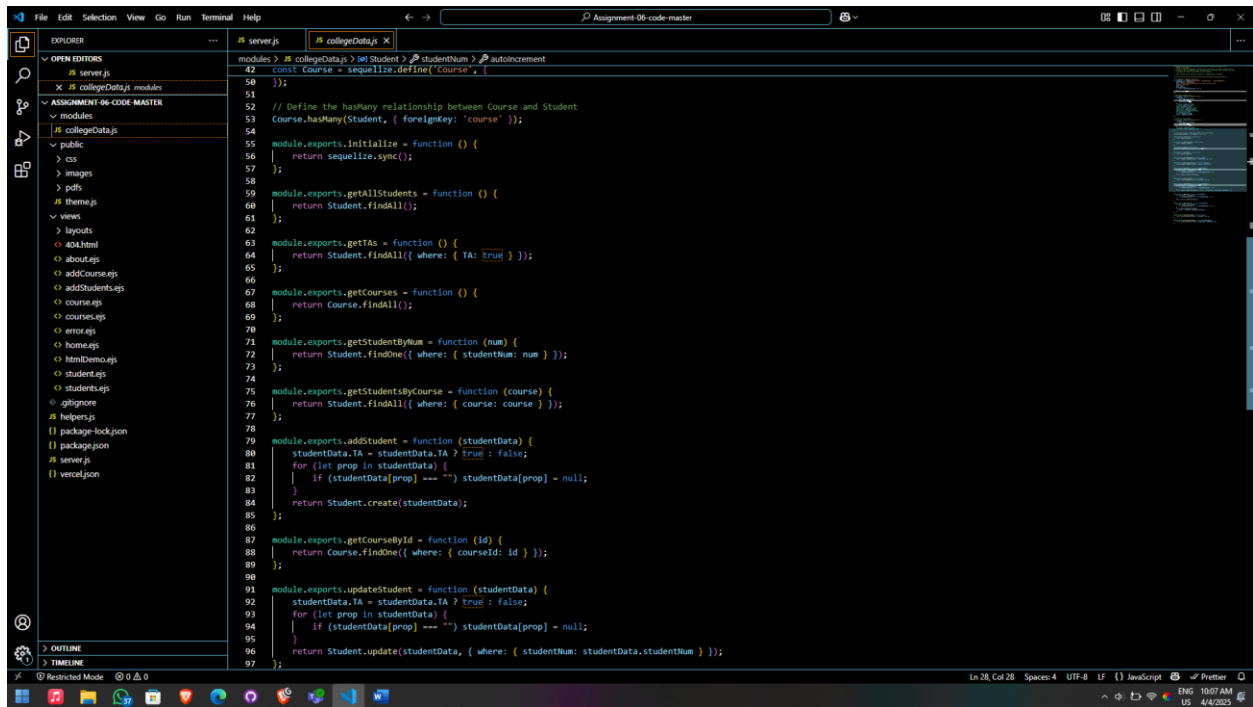
studentNum	firstName	lastName	email	addressStreet	addressCity	addressProvince		
1	Asintha Dilshan Jayasekara	Wisurumana Arachchige	crazyasin28@gmail.com	1485 Birchmount Road	Scarborough	ON		
2	Test	Test	test@com	1 test road	test city	test		
4	Tony	Stark	tony.sgmail.com	2 New York	New Jersey	US		

3 rows • 344ms

collegeData.js

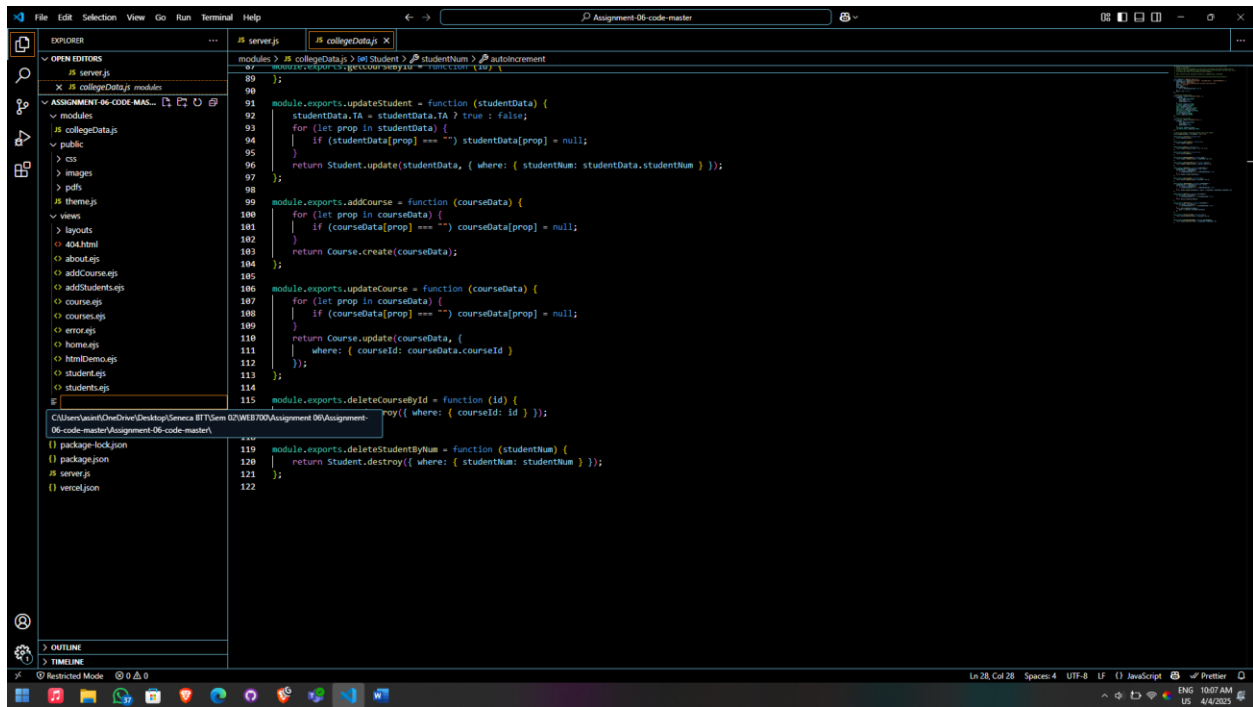
```
1 // WEB700 Assignment 2
2 * I declare that this assignment is my own work in accordance with Seneca Academic Policy.
3 * No part of this assignment has been copied manually or electronically from any other source
4 * (including web sites) or distributed to other students.
5
6 * Name: Asintha Dilshan Jayasekara Student ID: 170388235 Date: 16/02/2025
7
8
9
10
11 const Sequelize = require('sequelize');
12 var sequelize = new Sequelize('SenecaDB', 'npg_By3N3KULHx', {
13   dialectModule: require('pg'),
14   host: 'ep-still-band-a5dvdvj-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.getStudentsByNum = function (num) {
68   return Student.findOne({ where: { studentNum: num } });
69 };
70
71 module.exports.getStudentsByCourse = function (course) {
72   return Student.findAll({ where: { course: course } });
73 };
74
75 module.exports.addStudent = function (studentData) {
76   studentData.TA = studentData.TA ? true : false;
77   for (let prop in studentData) {
78     if (studentData[prop] === '') studentData[prop] = null;
79   }
80   return Student.create(studentData);
81 };
82
83 module.exports.getCourseById = function (id) {
84   return Course.findOne({ where: { courseId: id } });
85 };
86
87 module.exports.updateStudent = function (studentData) {
88   studentData.TA = studentData.TA ? true : false;
89   for (let prop in studentData) {
90     if (studentData[prop] === '') studentData[prop] = null;
91   }
92   return Student.update(studentData, { where: { studentNum: studentData.studentNum } });
93 };
94
95
96
97
```



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: { courseId: id } });
99 };
100
101 module.exports.deleteStudentByNum = function (studentNum) {
102   return Student.destroy({ where: { studentNum: studentNum } });
103 };
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
```

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)

[illegible]

The screenshot displays a web browser window with a form titled "addStudent.jsp". The form is rendered in a table layout with three columns. The first column contains a "Teaching Assistant" checkbox. The second column contains an "Enrollment Status" dropdown menu. The third column contains a "Course" dropdown menu. The browser's developer tools are open, showing the HTML structure of the form. The HTML code is as follows:

```
<table border="1">
  <tr>
    <td>
      <div class="checkbox">
        <input type="checkbox" value="true" id="taCheck"/>
        <label class="form-check-label" for="taCheck">Teaching Assistant</label>
      </div>
    </td>
    <td>
      <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>
    </td>
    <td>
      <div class="form-group">
        <label>Enrolled in Course</label>
        <div class="form-control" name="course" id="course" required>
          <select>
            <option value="">Select Course</option>
            <option value="1">1</option>
            <option value="2">2</option>
            <option value="3">3</option>
            <option value="4">4</option>
            <option value="5">5</option>
            <option value="6">6</option>
            <option value="7">7</option>
            <option value="8">8</option>
            <option value="9">9</option>
            <option value="10">10</option>
            <option value="11">11</option>
            <option value="12">12</option>
            <option value="13">13</option>
            <option value="14">14</option>
            <option value="15">15</option>
            <option value="16">16</option>
            <option value="17">17</option>
            <option value="18">18</option>
            <option value="19">19</option>
            <option value="20">20</option>
            <option value="21">21</option>
            <option value="22">22</option>
            <option value="23">23</option>
            <option value="24">24</option>
            <option value="25">25</option>
            <option value="26">26</option>
            <option value="27">27</option>
            <option value="28">28</option>
            <option value="29">29</option>
            <option value="30">30</option>
            <option value="31">31</option>
            <option value="32">32</option>
            <option value="33">33</option>
            <option value="34">34</option>
            <option value="35">35</option>
            <option value="36">36</option>
            <option value="37">37</option>
            <option value="38">38</option>
            <option value="39">39</option>
            <option value="40">40</option>
            <option value="41">41</option>
            <option value="42">42</option>
            <option value="43">43</option>
            <option value="44">44</option>
            <option value="45">45</option>
            <option value="46">46</option>
            <option value="47">47</option>
            <option value="48">48</option>
            <option value="49">49</option>
            <option value="50">50</option>
            <option value="51">51</option>
            <option value="52">52</option>
            <option value="53">53</option>
            <option value="54">54</option>
            <option value="55">55</option>
            <option value="56">56</option>
            <option value="57">57</option>
            <option value="58">58</option>
            <option value="59">59</option>
            <option value="60">60</option>
            <option value="61">61</option>
            <option value="62">62</option>
            <option value="63">63</option>
            <option value="64">64</option>
            <option value="65">65</option>
            <option value="66">66</option>
            <option value="67">67</option>
            <option value="68">68</option>
            <option value="69">69</option>
            <option value="70">70</option>
            <option value="71">71</option>
            <option value="72">72</option>
            <option value="73">73</option>
            <option value="74">74</option>
            <option value="75">75</option>
            <option value="76">76</option>
            <option value="77">77</option>
            <option value="78">78</option>
            <option value="79">79</option>
            <option value="80">80</option>
            <option value="81">81</option>
            <option value="82">82</option>
            <option value="83">83</option>
            <option value="84">84</option>
            <option value="85">85</option>
            <option value="86">86</option>
            <option value="87">87</option>
            <option value="88">88</option>
            <option value="89">89</option>
            <option value="90">90</option>
            <option value="91">91</option>
            <option value="92">92</option>
            <option value="93">93</option>
            <option value="94">94</option>
            <option value="95">95</option>
            <option value="96">96</option>
            <option value="97">97</option>
            <option value="98">98</option>
            <option value="99">99</option>
            <option value="100">100</option>
            <option value="101">101</option>
            <option value="102">102</option>
            <option value="103">103</option>
            <option value="104">104</option>
            <option value="105">105</option>
            <option value="106">106</option>
            <option value="107">107</option>
            <option value="108">108</option>
            <option value="109">109</option>
            <option value="110">110</option>
            <option value="111">111</option>
            <option value="112">112</option>
            <option value="113">113</option>
            <option value="114">114</option>
            <option value="115">115</option>
            <option value="116">116</option>
            <option value="117">117</option>
            <option value="118">118</option>
            <option value="119">119</option>
            <option value="120">120</option>
            <option value="121">121</option>
            <option value="122">122</option>
            <option value="123">123</option>
            <option value="124">124</option>
            <option value="125">125</option>
            <option value="126">126</option>
            <option value="127">127</option>
            <option value="128">128</option>
            <option value="129">129</option>
            <option value="130">130</option>
            <option value="131">131</option>
            <option value="132">132</option>
            <option value="133">133</option>
            <option value="134">134</option>
            <option value="135">135</option>
            <option value="136">136</option>
            <option value="137">137</option>
            <option value="138">138</option>
            <option value="139">139</option>
            <option value="140">140</option>
            <option value="141">141</option>
            <option value="142">142</option>
            <option value="143">143</option>
            <option value="144">144</option>
            <option value="145">145</option>
            <option value="146">146</option>
            <option value="147">147</option>
            <option value="148">148</option>
            <option value="149">149</option>
            <option value="150">150</option>
            <option value="151">151</option>
            <option value="152">152</option>
            <option value="153">153</option>
            <option value="154">154</option>
            <option value="155">155</option>
            <option value="156">156</option>
            <option value="157">157</option>
            <option value="158">158</option>
            <option value="159">159</option>
            <option value="160">160</option>
            <option value="161">161</option>
            <option value="162">162</option>
            <option value="163">163</option>
            <option value="164">164</option>
            <option value="165">165</option>
            <option value="166">166</option>
            <option value="167">167</option>
            <option value="168">168</option>
            <option
```

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>
26 </div>
```

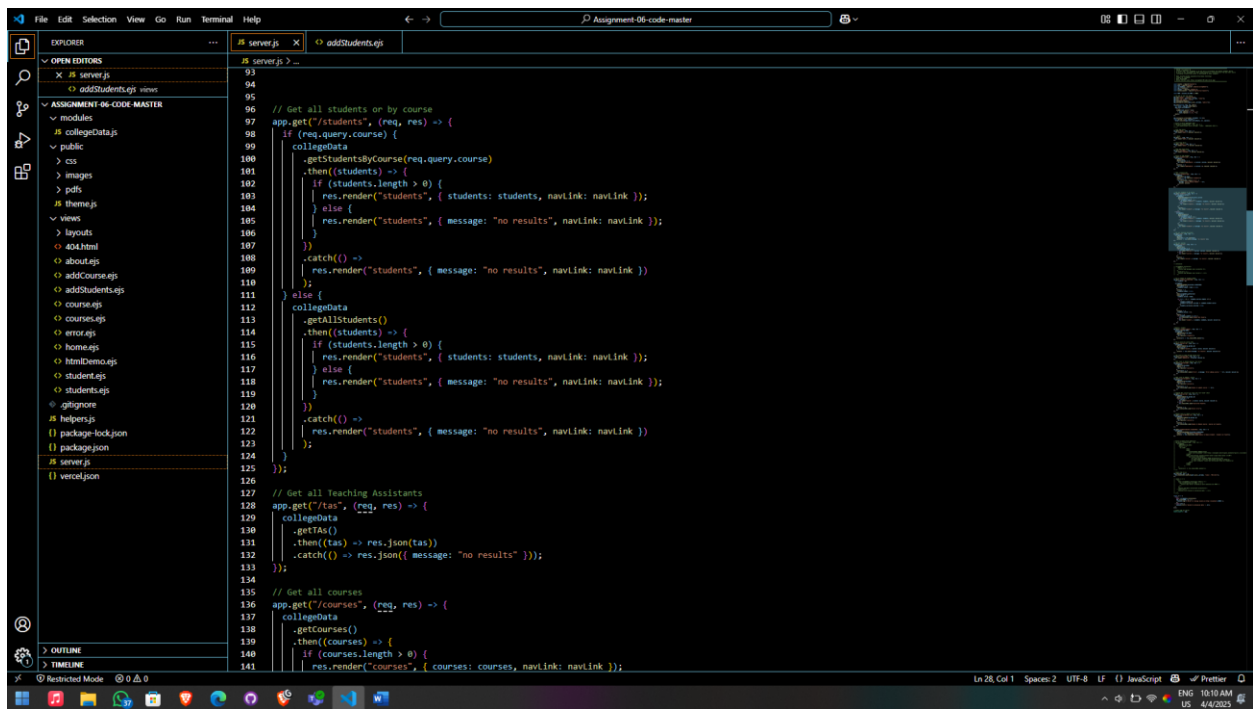
WEB700_Assignment6_Asintha Dilshan Jayasekara (170388235)

Server.js

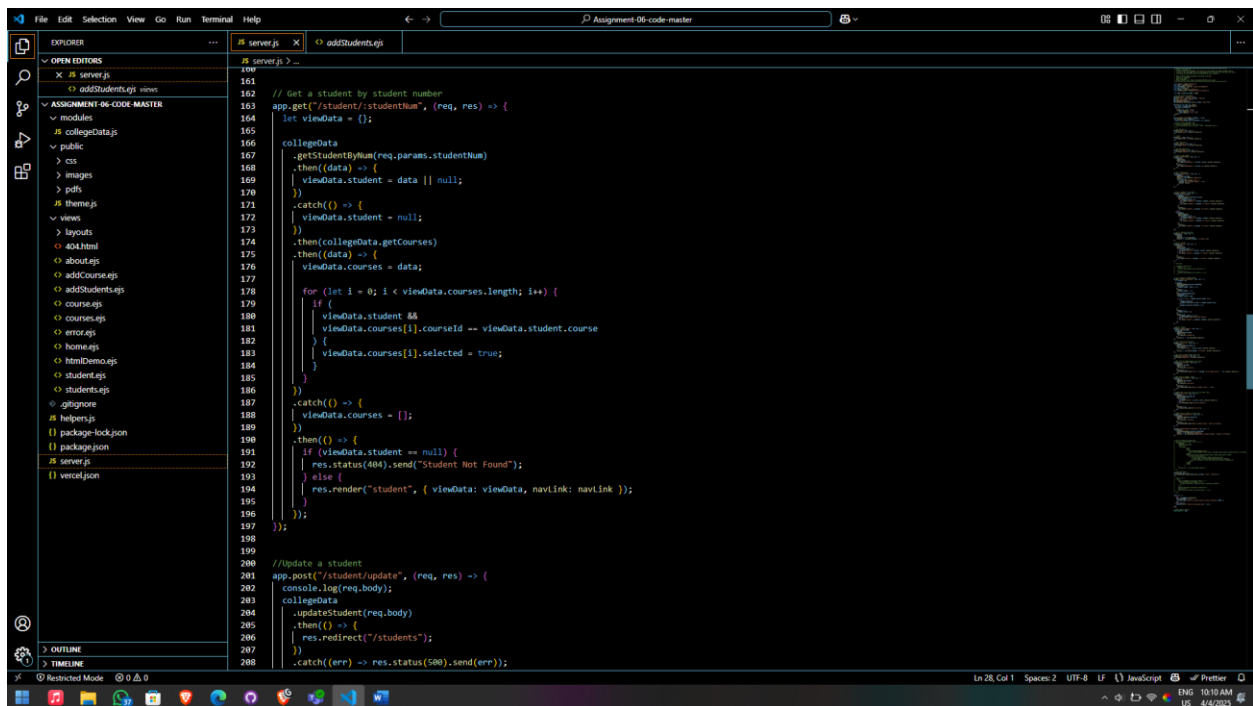
```
1 // =====
2 * WEB700 Assignment 06
3 * I declare that this assignment is my own work in accordance with Seneca Academic Policy.
4 * No part of this assignment has been copied manually or electronically from any other source
5 * (including 3rd party web sites) or distributed to other students.
6 *
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 app.use(expressLayouts);
28
29 app.use(function (req, res, next) {
30   let route = req.path.substring(1);
31   app.locals.activeRoute =
32     "/" +
33     (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 // Route to Serve addStudent.html
44 // app.get("/students/add", (req, res) => {
45 //   res.sendFile(path.join(__dirname, "views", "addStudent.html"));
46 // });
47
48 // Home Page Route
49 app.get("/", (req, res) => {
```

```
46 // });
47
48 // Home Page Route
49 app.get("/", (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("/teaching-assistants", (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)

The image shows a VS Code editor window with a file explorer on the left and a code editor in the center. The file explorer shows a project structure with folders like 'views', 'public', 'controllers', and 'models'. The code editor displays the 'server.js' file, which contains the application logic. The code includes routes for GET, POST, and PUT methods, handling course data and user authentication. The status bar at the bottom indicates the file is located at 'Ln 28, Col 1' and the workspace is 'Assignment-06-master'.

The screenshot shows a VS Code editor with a Node.js application. The Explorer sidebar on the left displays the file structure, including folders for 'modules', 'public', 'images', 'pdfs', 'theme.js', and 'views'. The main editor displays the 'addStudents.js' file, which contains routes for adding, deleting, and updating students. The code uses the 'mongoose' library for database operations. The bottom status bar shows the file is at line 28, column 1, with 2 spaces and 2 tabs. The language mode is set to JavaScript.

WEB700_Assignment6_Asintha Dilshan Jayasekara (170388235)

```
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
```