

# WEB700 Assignment 5

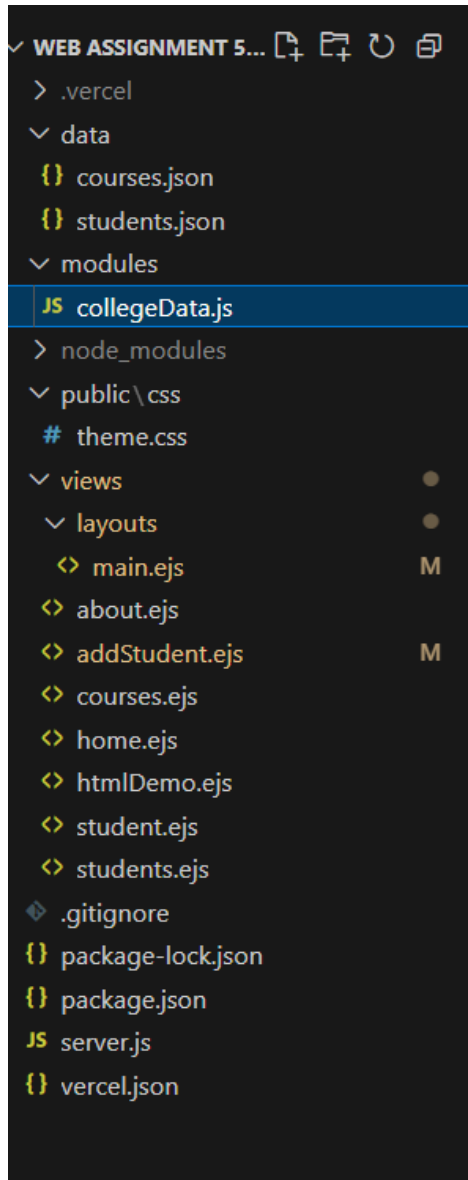
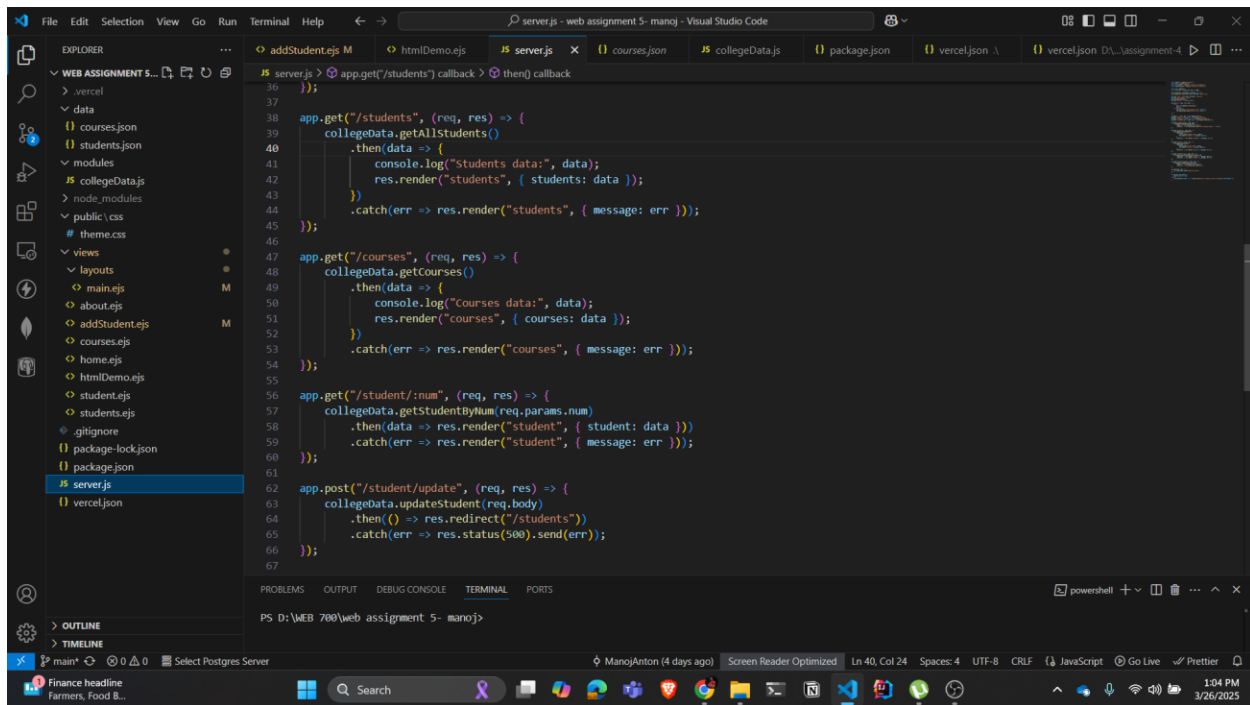


Figure 1: Project Structure

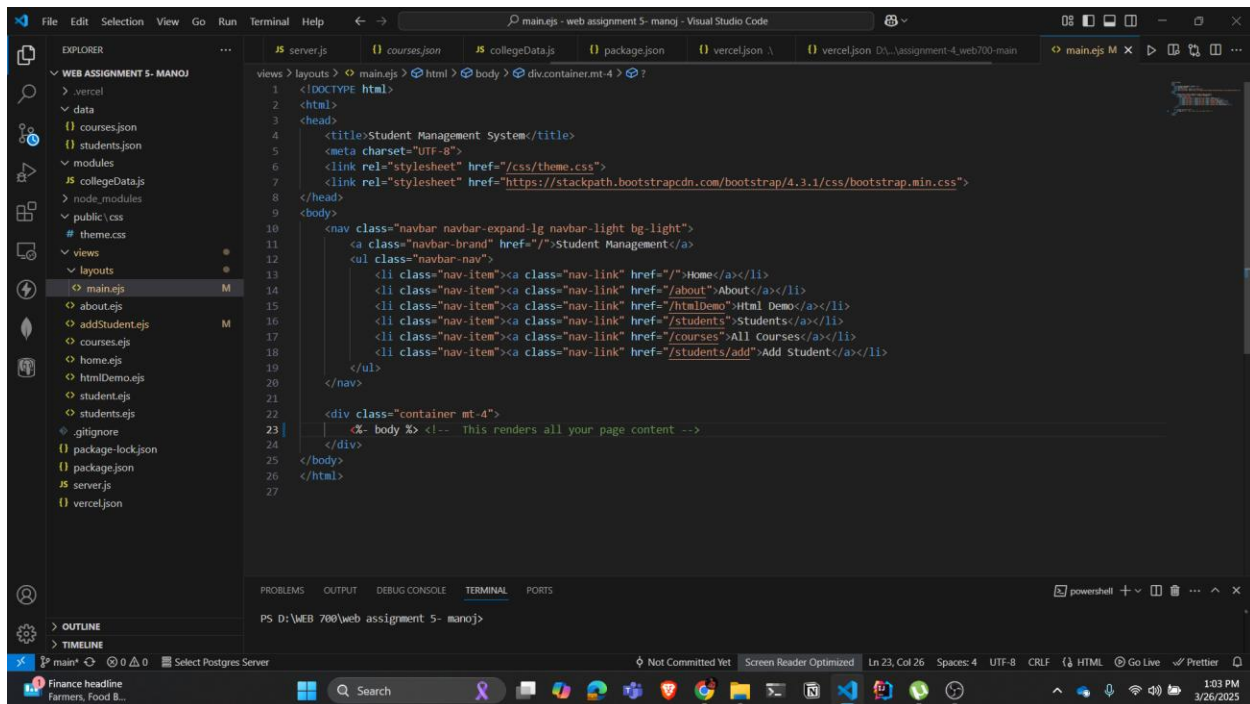


The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying a project structure for 'WEB ASSIGNMENT 5- MANOJ'. The file explorer includes folders like 'data', 'modules', 'node\_modules', 'public', 'views', and 'layouts', and files like 'courses.json', 'students.json', 'collegeData.js', 'package-lock.json', 'package.json', 'server.js', 'vercel.json', 'theme.css', 'main.ejs', 'about.ejs', 'addStudent.ejs', 'courses.ejs', 'home.ejs', 'htmlDemo.ejs', 'student.ejs', 'students.ejs', and 'gillignore'. The main editor window displays the 'server.js' file with the following code:

```
server.js > app.get("/students") callback > then() callback
36
37
38 app.get("/students", (req, res) => {
39   collegeData.getAllStudents()
40   .then(data => {
41     console.log("Students data:", data);
42     res.render("students", { students: data });
43   })
44   .catch(err => res.render("students", { message: err }));
45 });
46
47 app.get("/courses", (req, res) => {
48   collegeData.getCourses()
49   .then(data => {
50     console.log("Courses data:", data);
51     res.render("courses", { courses: data });
52   })
53   .catch(err => res.render("courses", { message: err }));
54 });
55
56 app.get("/student/:num", (req, res) => {
57   collegeData.getStudentByNum(req.params.num)
58   .then(data => res.render("student", { student: data }))
59   .catch(err => res.render("student", { message: err }));
60 });
61
62 app.post("/student/update", (req, res) => {
63   collegeData.updateStudent(req.body)
64   .then(() => res.redirect("/students"))
65   .catch(err => res.status(500).send(err));
66 });
67
```

The terminal at the bottom shows the command 'PS D:\WEB 700\web assignment 5- manoj>'.

Figure 2:Server.js

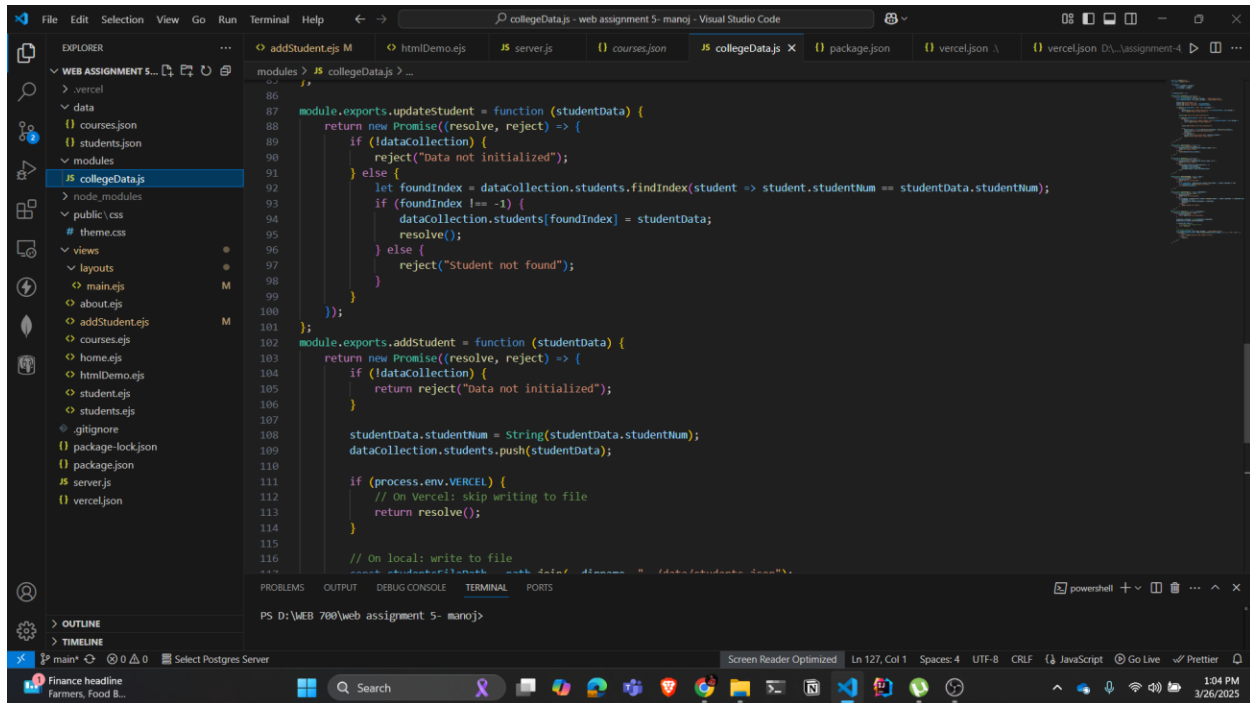


The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying the same project structure as Figure 2. The main editor window displays the 'main.ejs' file with the following code:

```
views > layouts > main.ejs > html > body > div.container.mt-4 > ?
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Student Management System</title>
5   <meta charset="UTF-8">
6   <link rel="stylesheet" href="/css/theme.css">
7   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
8 </head>
9 <body>
10   <nav class="navbar navbar-expand-lg navbar-light bg-light">
11     <a class="navbar-brand" href="/">Student Management</a>
12     <ul class="navbar-nav">
13       <li class="nav-item"><a class="nav-link" href="/">Home</a></li>
14       <li class="nav-item"><a class="nav-link" href="/about">About</a></li>
15       <li class="nav-item"><a class="nav-link" href="/htmlDemo">Html Demo</a></li>
16       <li class="nav-item"><a class="nav-link" href="/students">Students</a></li>
17       <li class="nav-item"><a class="nav-link" href="/courses">All courses</a></li>
18       <li class="nav-item"><a class="nav-link" href="/students/add">Add Student</a></li>
19     </ul>
20   </nav>
21
22   <div class="container mt-4">
23     <!-- body --> <!-- This renders all your page content -->
24   </div>
25 </body>
26 </html>
27
```

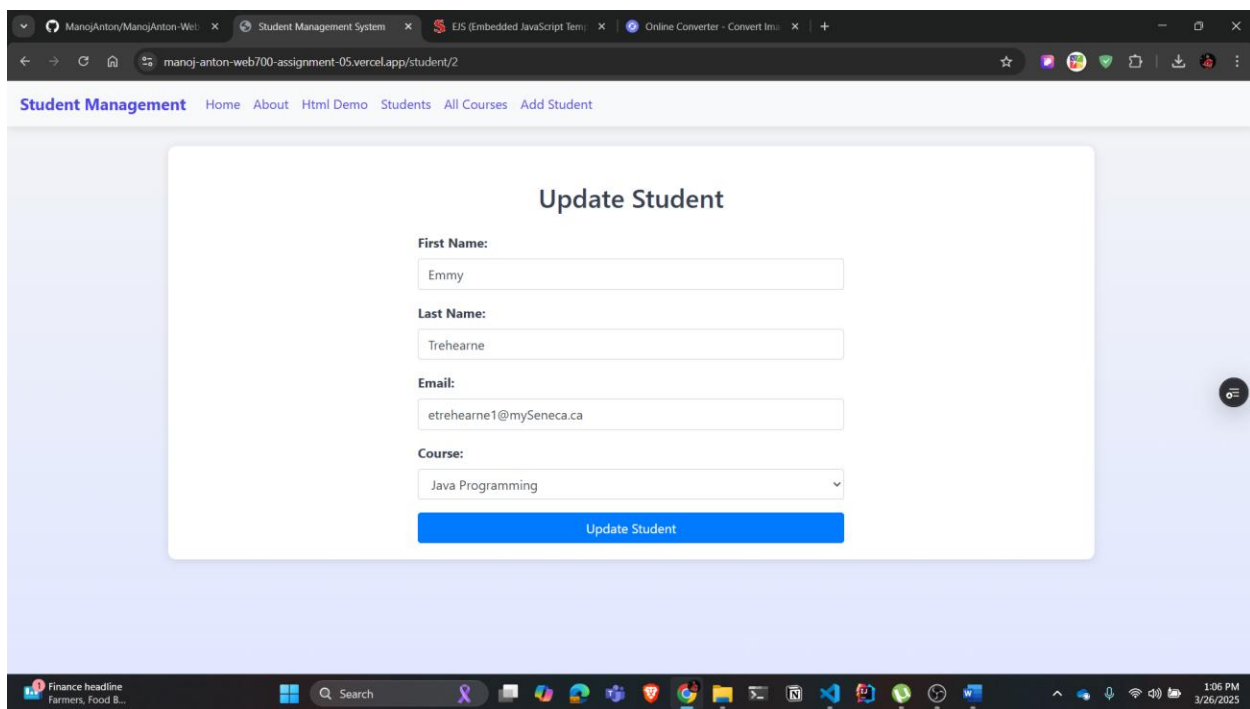
The terminal at the bottom shows the command 'PS D:\WEB 700\web assignment 5- manoj>'.

Figure 3:Main.ejs



```
86 // ...
87 module.exports.updateStudent = function (studentData) {
88   return new Promise((resolve, reject) => {
89     if (!dataCollection) {
90       reject("data not initialized");
91     } else {
92       let foundIndex = dataCollection.students.findIndex(student => student.studentNum == studentData.studentNum);
93       if (foundIndex !== -1) {
94         dataCollection.students[foundIndex] = studentData;
95         resolve();
96       } else {
97         reject("Student not found");
98       }
99     }
100   });
101 };
102 module.exports.addStudent = function (studentData) {
103   return new Promise((resolve, reject) => {
104     if (!dataCollection) {
105       return reject("data not initialized");
106     }
107     studentData.studentNum = String(studentData.studentNum);
108     dataCollection.students.push(studentData);
109     if (process.env.VERCEL) {
110       // On Vercel: skip writing to file
111       return resolve();
112     }
113     // On local: write to file
114     // ...
115   });
116 };
```

Figure 4:College data



Student Management | Home | About | Html Demo | Students | All Courses | Add Student

### Update Student

First Name:

Last Name:

Email:

Course:

Figure 5:

The screenshot shows a web browser window with the URL `manoj-anton-web700-assignment-05.vercelapp/students/add`. The page has a navigation bar with links: [Student Management](#), [Home](#), [About](#), [Html Demo](#), [Students](#), [All Courses](#), and [Add Student](#). The main content area is titled "Add Student" and contains two sections: "Personal Information" and "Academic Information".

**Personal Information**

Student Number:

First Name:

Last Name:

Email:

**Academic Information**

Course:

Figure 6

The screenshot shows a web browser window with the URL `manoj-anton-web700-assignment-05.vercelapp/courses`. The page has a navigation bar with links: [Student Management](#), [Home](#), [About](#), [Html Demo](#), [Students](#), [All Courses](#), and [Add Student](#). The main content area is titled "Courses" and displays a table of courses.

Course ID	Course Code	Course Description
1	DES720	Relational Database Design and Implementation
2	JAV745	Java Programming
3	OPS705	Introduction to Cloud Computing
4	SQL710	Database Administration and Management
5	WEB700	Web Programming
6	CAP805	Applied Capstone Project
7	CJV805	Database Connectivity Using Java
8	DBD800	Accessing Big Data
9	DBW825	Datawarehousing
10	SEC835	Security in Databases and Web Applications

Figure 7

Student Management System

Home About Html Demo Students All Courses Add Student

### Students

Student Number	Full Name	Email	Course
1	Foster Thorburn	fthorburn0@mySeneca.ca	2
2	Emmy Trehearne	etrehearne1@mySeneca.ca	2
3	Zonnya Laytham	zlaytham2@mySeneca.ca	2
4	Asia Bollon	abollon3@mySeneca.ca	4
5	Ysabel Collyns	ycollyns4@mySeneca.ca	6
6	Tremain Cassy	tcassy5@mySeneca.ca	7
7	Jess Jago	jjago6@mySeneca.ca	6
8	Rodi Tant	rtant7@mySeneca.ca	5
9	Henri Rikard	hrikard8@mySeneca.ca	1
10	Beatrisa Wanne	bwanne9@mySeneca.ca	3

Figure 8