# LABORATORY EXERCISE 3

# ROUTING AND MVC STRUCTURE

**Learning Objectives**

By the end of this laboratory exercise, students should be able to:

* Understand CodeIgniter's MVC (Model-View-Controller) architecture and routing system.
* Configure custom routes in `app/Config/Routes.php`.
* Create a controller with multiple methods and their corresponding views.
* Implement navigation between pages in a CodeIgniter application.
* Utilize GitHub to track and manage project versions.

**Prerequisite student experiences and knowledge**

Before starting this exercise, students should have:

* Basic knowledge of HTML, CSS, and PHP.
* Familiarity with CodeIgniter's file structure.
* Experience using a local web server (XAMPP/WAMP/LAMP).
* Understanding of basic Git commands and GitHub usage.
* Ability to use a text editor/IDE such as Visual Studio Code.

**Background**

CodeIgniter follows the **MVC architecture**, separating application logic into Models, Views, and Controllers. Routing in CodeIgniter determines which controller and method are executed based on the URL. Understanding how to set up routes and navigate between pages is essential for building dynamic and maintainable applications. GitHub will be used for version control and documenting each stage of development.

**Materials/Resources**

* **Personal Computer with Internet Access**
* **XAMPP/WAMP/LAMP server installed**
* **CodeIgniter Framework (latest version)**
* **Visual Studio Code or any code editor**
* **Git and GitHub Account**
* **Web Browser (Chrome, Firefox, etc.)**

**Laboratory Activity**

**Step 1: Project Setup**

1. Open your previously created CodeIgniter project **ITE311-LASTNAME**.
2. Run the local server and confirm the project is working: **http://localhost/ITE311-LASTNAME**

**Step 2: Create a Home Controller**

1. Navigate to **app/Controllers/** and create a file named **Home.php**.
2. Add the following methods:
   * index() → Loads a homepage.
   * about() → Loads an about page.
   * contact() → Loads a contact page.

**Step 3: Configure Routes**

1. Open `app/Config/Routes.php`.
2. Add custom routes for:
   * `/` → `Home::index`
   * `/about` → `Home::about`
   * `/contact` → `Home::contact`

**Step 4: Create Views**

1. Inside `app/Views/`, create:
   * `index.php` (Homepage)
   * `about.php` (About Page)
   * `contact.php` (Contact Page)
2. Add navigation links between the pages in each view using HTML anchor tags.

**Step 5: Test Navigation**

1. Run the application and test:
   * **Homepage → About → Contact**
2. Ensure that the navigation works correctly and each page displays its content.

**Step 6: Push to GitHub**

1. Stage and commit your migration and seeder files:

**git add .**

**git commit -m "Added Home controller, routes, and views for basic navigation"**

Output / Results

* Screenshot of the homepage, about page, and contact page.
* Screenshot of `**Routes.php**` configuration.
* A screenshot of the GitHub repository with the latest commit is attached.

**QUESTIONS:**

1. Explain the role of the controller in CodeIgniter's MVC structure.

The role of the controller in CodeIgniter's MVC structure is to act as the middleman that handles user requests and coordinates between models and views. When I search the <http://localhost/ITE311-ALEJADO/public/about>, the controller receives this request, processes logic and then loads the appropriate view to display the about page.

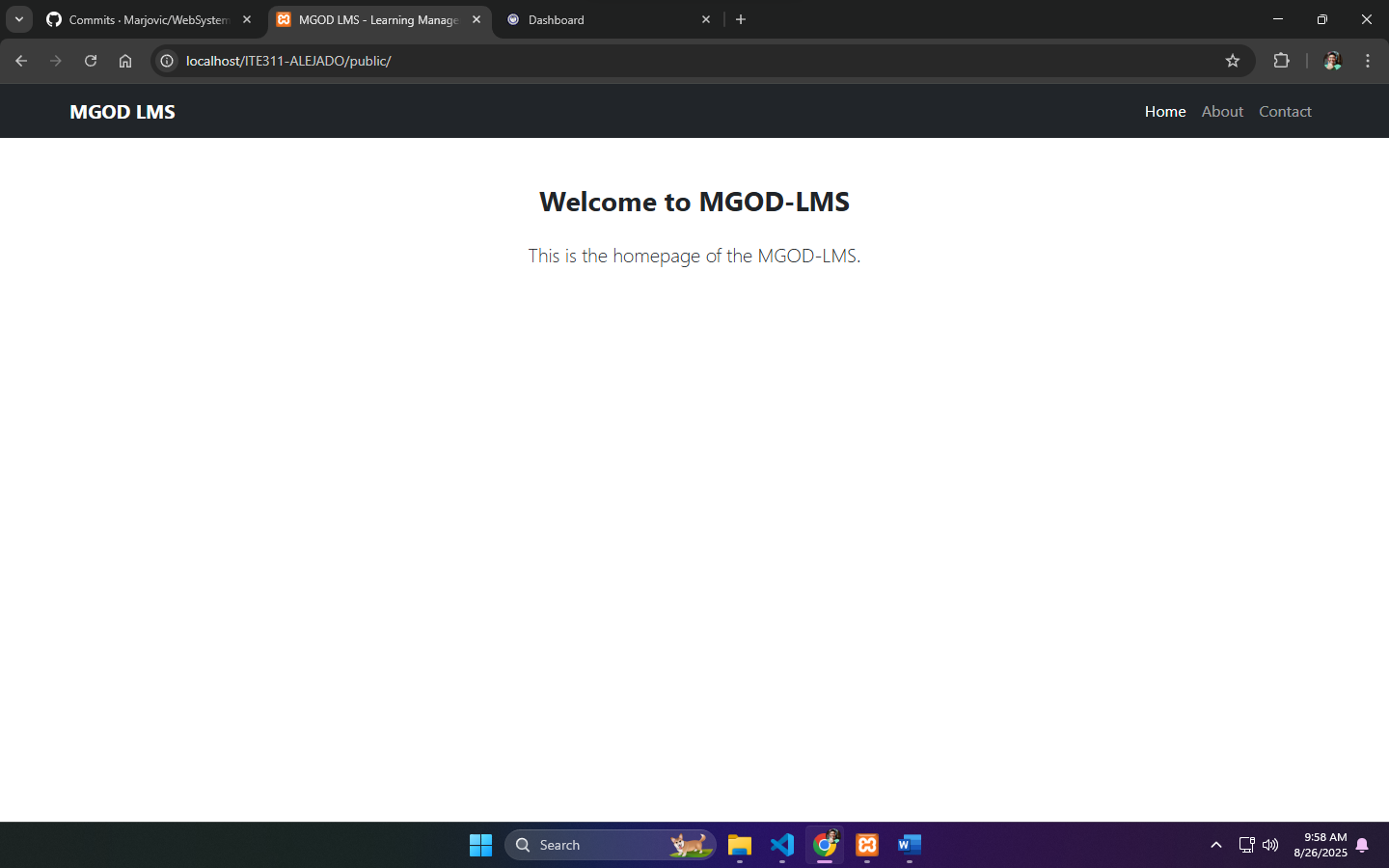
1. Why is routing important in a web application?

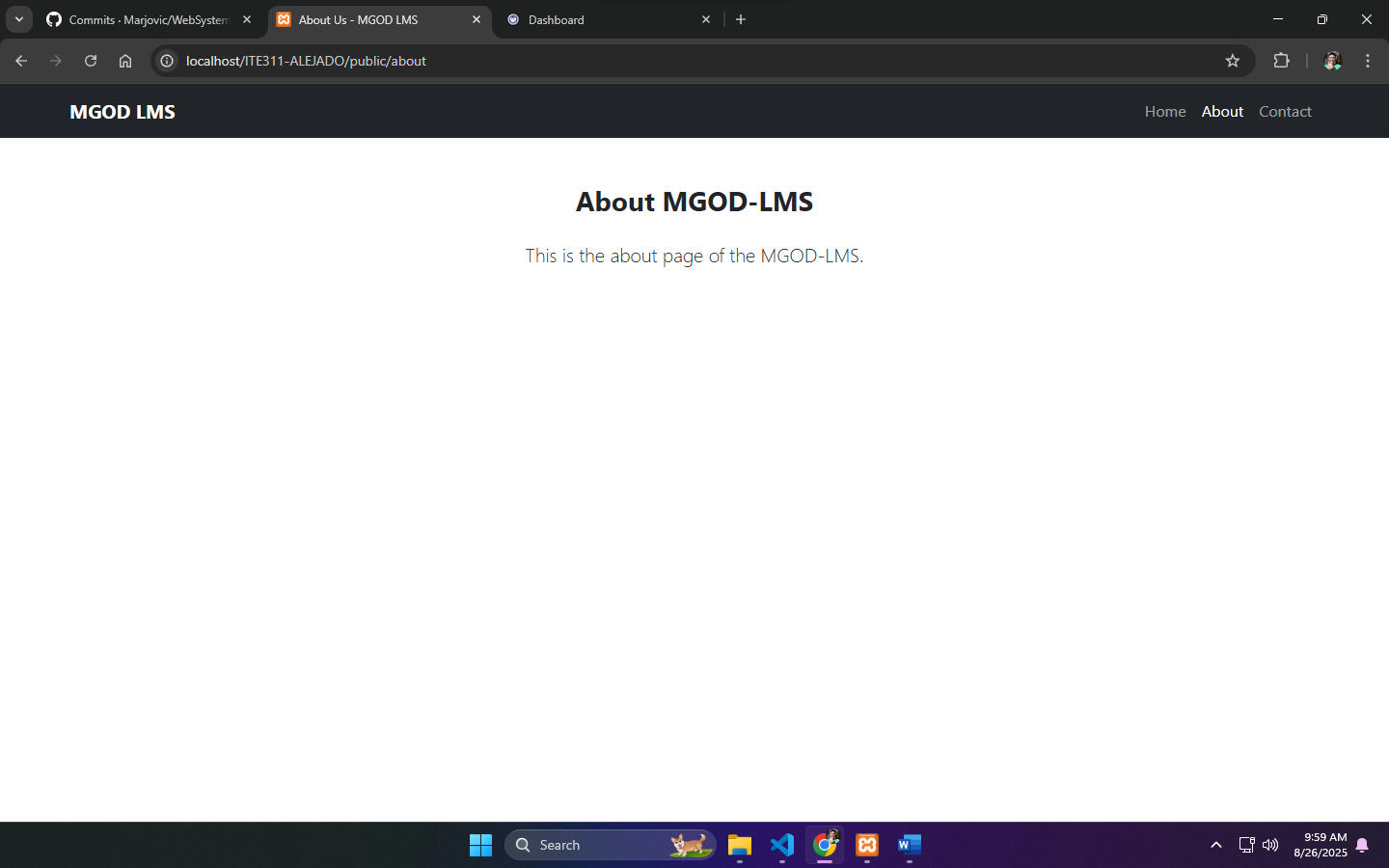
Routing is important in a web application because it connects each URL to the right part of code. For example, when I search <http://localhost/ITE311-ALEJADO/public/about>, routing sends the request to Home::about so the correct page shows. It keeps links clean and lets me control which pages exist and what code runs for each navigation.

1. How does GitHub help in tracking changes in your CodeIgniter project?

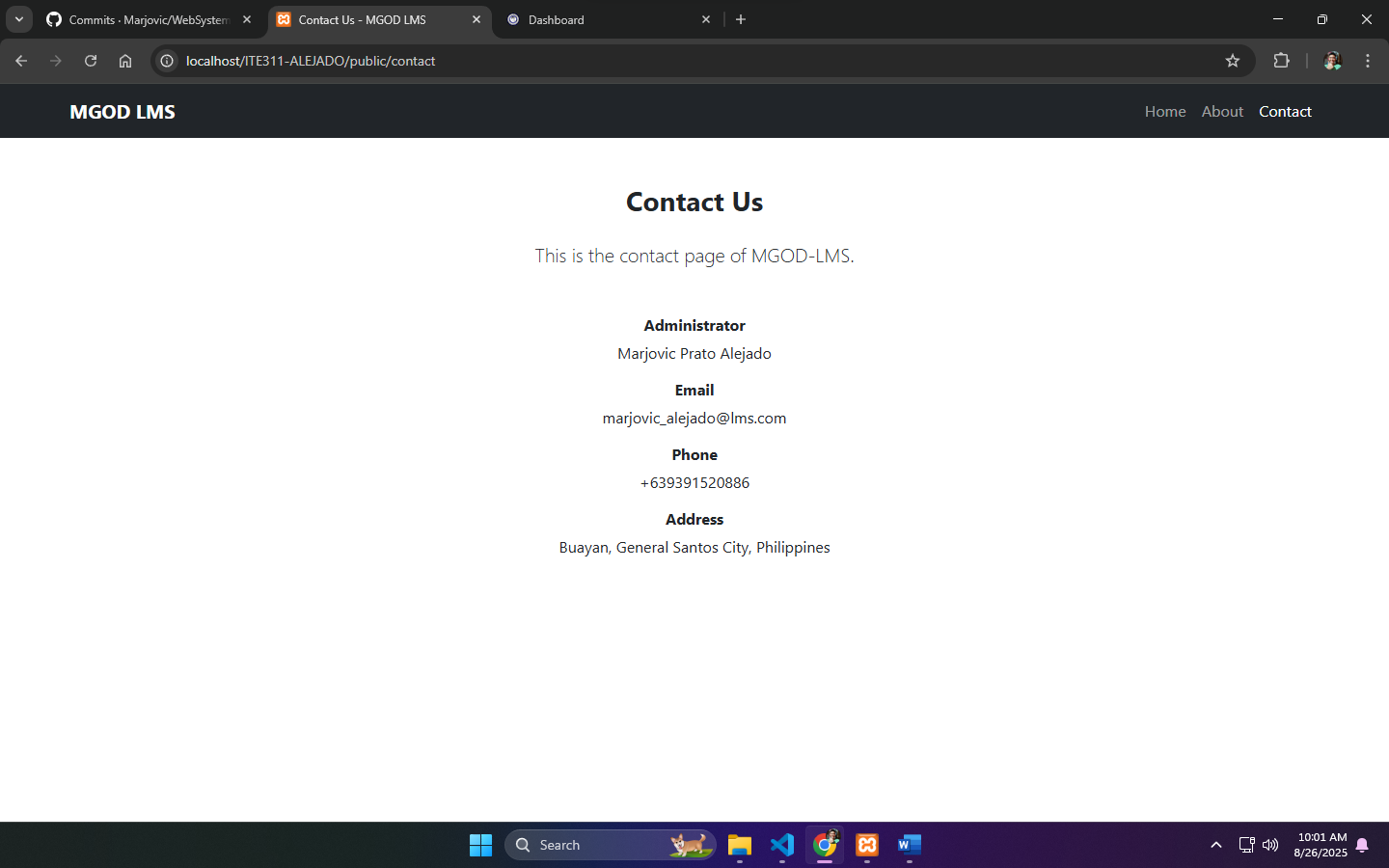
Github helps in tracking changes in my CodeIgniter project through commits and a clear history of what changed. I add and commit with a message, push to GitHub, and it shows the changes, and when they happened. If I have mistakes, I can compare versions, and can go back to an older commit.

**Output / Results**

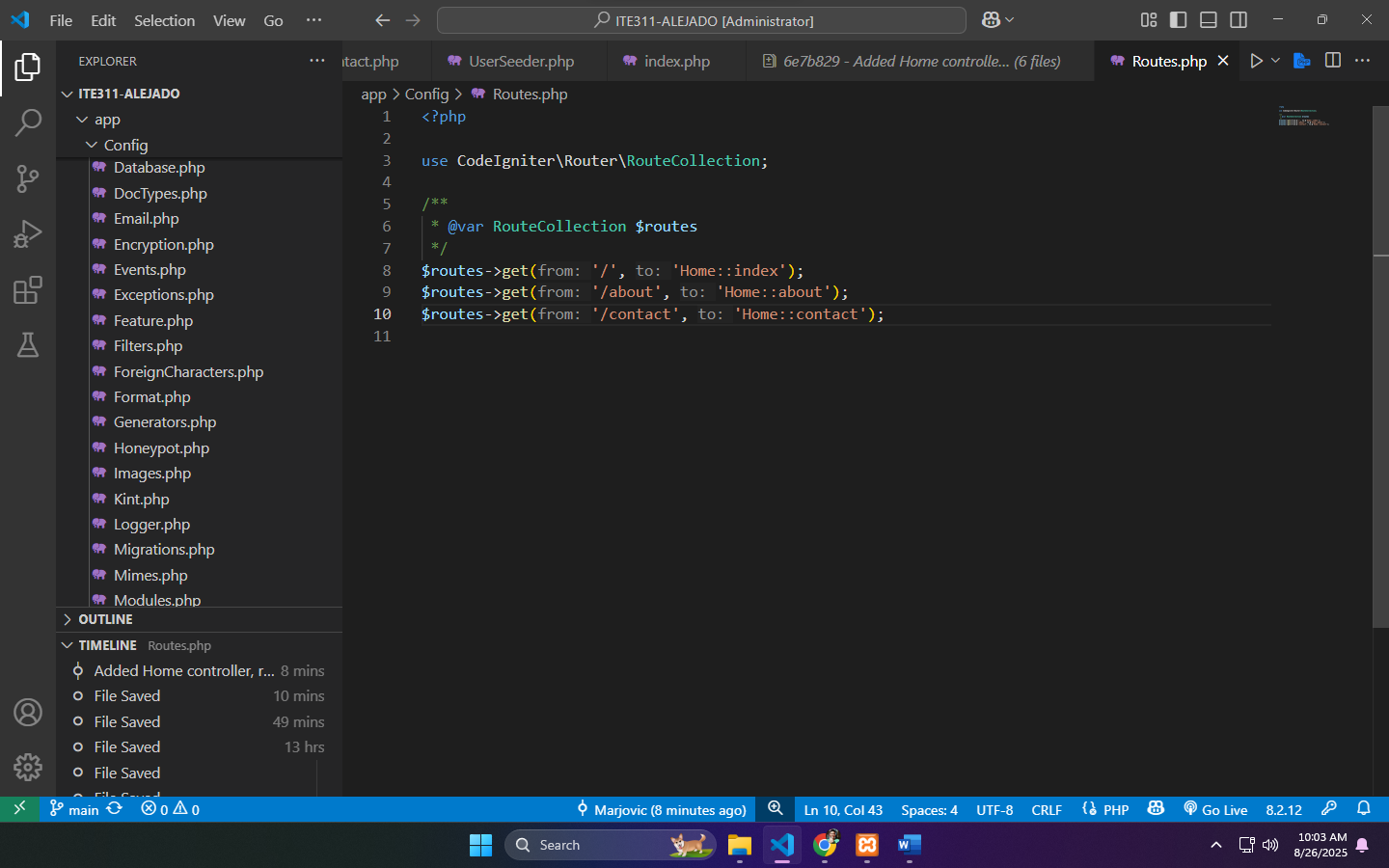
**Home Page**  
  


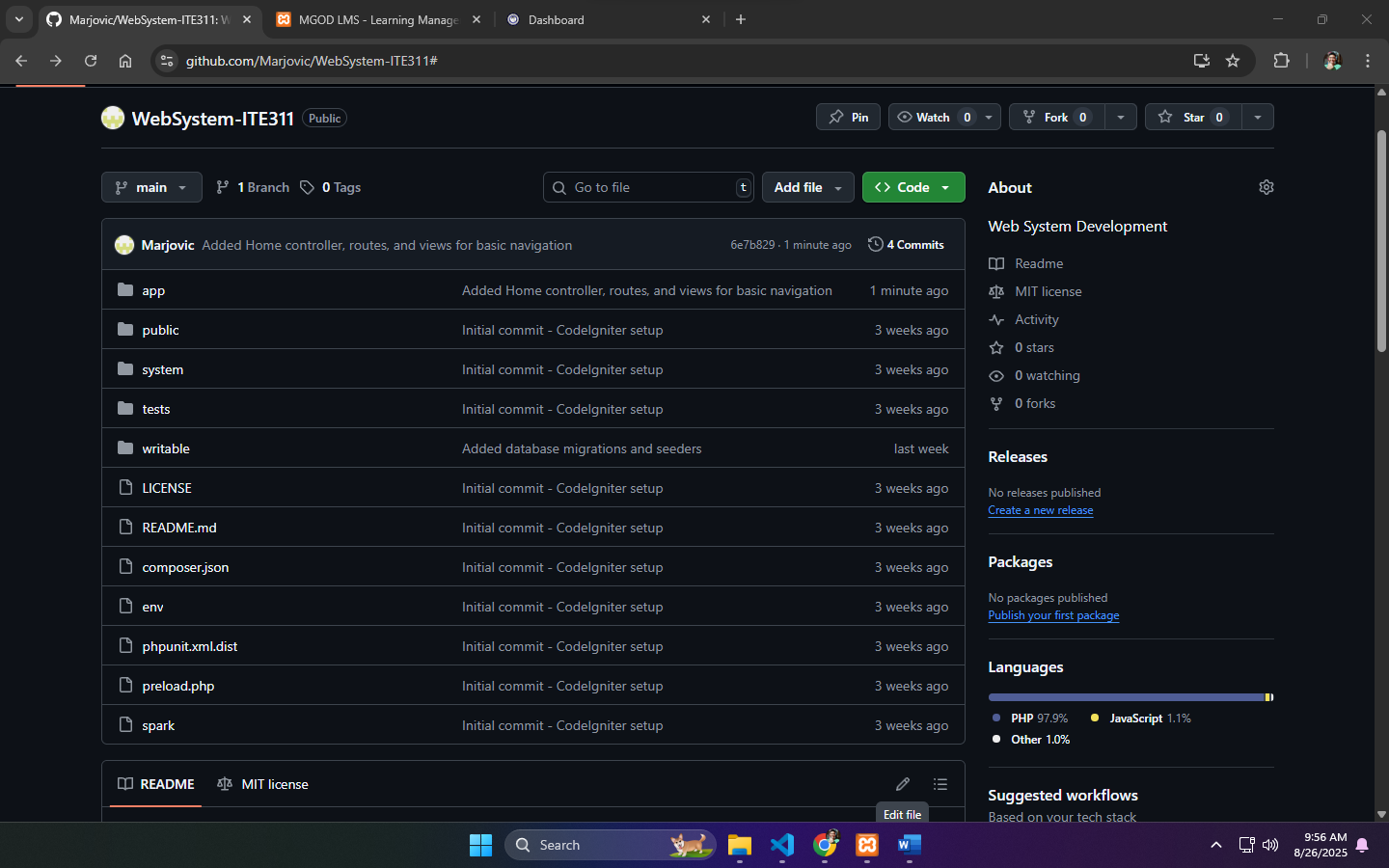
**About Page**   
  


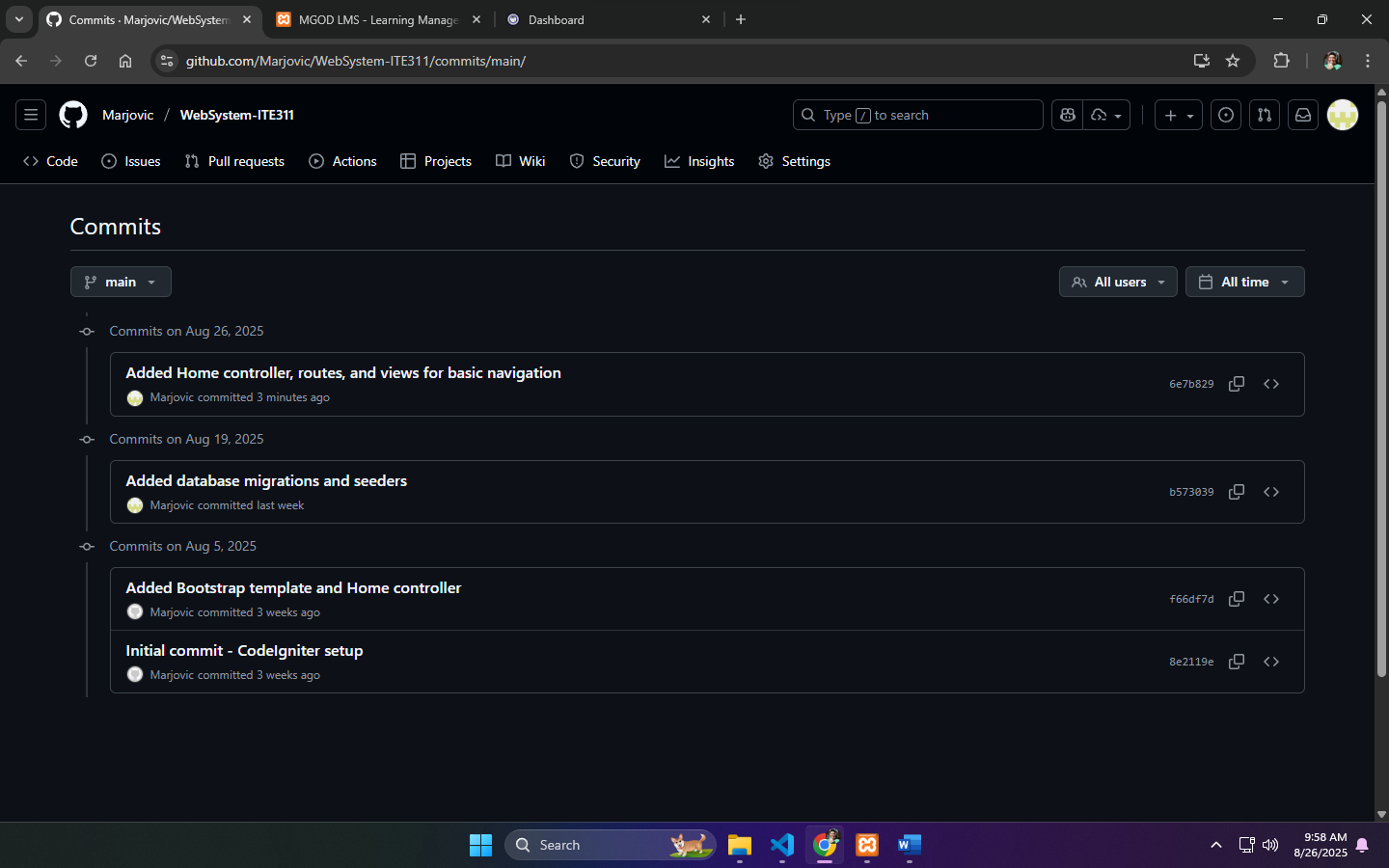
**Contact Page**



**app/Config/Routes.php**



GitHub Link: <https://github.com/Marjovic/WebSystem-ITE311>



**Conclusion**

In conclusion, Thanks to this Laboratory Exercise 3, I learned how Model, View, and Controller really is and how routing connects URLs to controller methods. I set custom routes in app/Config/Routes.php, I also made a Home controller with several methods, home, about and contact page views and added navigations. I also used GitHub to commit and push changes, track history, and manage versions safely.