COMP 2139 – Assignment 2

Due Date: Tuesday, April 16th, 2024 (11:59 pm)
Team Size: Maximum of **3 students** per group.
Project Name: **GBC_Travel-Group**-<*Number*>

Objective:
For this assignment, you will enhance your travel booking platform developed in Assignment 1 by integrating advanced features and functionalities focusing on security, architecture, error handling, and testing. The primary focus will be on implementing ASP.NET Core Identity for robust authentication and authorization.

Requirement:
The compulsory technologies required for this project are the following:

- **ASP.NET Core MVC**
- **C#**
- **Web Server (ex. IIS Express)**
- **Database (ex SQL Express)**
- **>= .NET6.0**
- **MVC Design**

The **minimum** project requirements have been summarized below.

Deliverables:
Please refer to the assignment requirements document "**Submission Checklist**".

Project Requirements (Compulsory Core Enhancements and Features)

Enhanced Identity Core Implementation:

1. **Implement ASP.NET Core Identity**

Objective: Integrate ASP.NET Core Identity into your GBC Travel Application for comprehensive user and role management.

Details:
- Implement user registration and management. Users should be able to sign up as either regular users (**travelers**) or **admin** users.
- Admin users are responsible for managing listings such as flights, hotels, and car rentals.
    - As a consequence, admin users (and only admin users) will need the necessary interfaces and views to manage all these functionalities.
- Regular users (travelers) should be able to search, book, and review travel services.

2. **Authentication:**

Objective: Establish and design a secure authentication system to protect user accounts and personal information.

Details:
- Implement login and logout functionalities.
- Ensure the registration process includes email verification for user authenticity.
- Secure user passwords using strong hashing algorithms provided by ASP.NET Core Identity.
- Implement a "forgot password" and “password reset” feature to enhance user account security.

3. **Authorization:**

Objective: Define and enforce access controls within the GBC Travel Application based on user roles.

Details:
- Define at least two roles: **Admin** and Regular User (**Traveler**).
- Admin users should have the authority to add, edit, or delete listings, and view all bookings.
- Regular users should be able to view and book travel services but should not have access to manage listings.
- Use role-based authorization to protect routes and functionalities accordingly, ensuring users can only access appropriate areas of the application.

4. **Customization:**

Objective: Tailor the ASP.NET Core Identity system to meet the specific needs of the GBC Travel Application.

Details:
- Customize user profiles to include additional information relevant to travel bookings, such as contact information, preferences, and profile pictures.
- Modify the login and registration pages to align with the GBC Travel Application's branding and user interface design.
- Extend the Identity model to support custom fields necessary for your travel application, such as frequent flyer numbers or hotel loyalty program IDs for travelers.

Advanced MVC Concepts Integration:

1. **Logging Filter:**

Objective: Implement a custom filter in your Travel Application to enhance logging.

Details:
- **Logging Filter**: Create a custom action filter that logs details of user activities and system events, such as search queries and booking attempts. This will help in monitoring user interactions and system performance.

2. **Custom Middleware:**

Objective: Develop custom middleware for the Travel Application to handle cross-cutting concerns such as logging and error handling more effectively.

Details:
- **Logging Middleware**: Implement middleware that logs incoming requests and their response status. This can help in tracking the flow of requests and identifying problematic areas in the application.
- **Error Handling Middleware**: Create middleware that catches and logs exceptions globally. Ensure that this middleware provides a mechanism to display custom error pages depending on the error type or HTTP status code.

3. **Advanced Routing:**

Objective: Implement advanced routing techniques in the GBC Travel Application to enhance the user experience and application maintainability.

Details:
- **Attribute Routing**: Use attribute routing to define more descriptive and intuitive routes. For example, configuring route templates for specific travel services like /flights/{id} for flight details or /hotels/{name} for hotel information.
- **Route Constraints**: Apply route constraints to validate route data directly within the URL, ensuring that users are navigating to valid endpoints.

4. **Error Handling:**

Objective: Establish a robust error handling framework within the GBC Travel Application to manage and respond to runtime errors gracefully.

Details:

- **Global Error Handling**: Configure global error handling in the application's Program.cs class to catch and log all unhandled exceptions. Ensure users are redirected to a custom error page that provides a friendly error message without exposing sensitive information.

- **Try-Catch Blocks**: Implement try-catch blocks in critical sections of your code, especially where external API calls or database transactions occur, to handle exceptions locally and provide contextual feedback to users.

- **Custom Error Pages**: Create custom error pages for different HTTP status codes (404 Not Found, 500 Internal Server Error, etc.) that align with the application's design and provide helpful information to the user on what to do next.

- **Logging**: Integrate detailed error logging using a framework like Serilog or NLog. Ensure that logs contain sufficient information to diagnose issues, including the timestamp, error message, stack trace, and user details when available

Unit Testing and Best Practices:

1. **Unit Testing:**

Objective: Develop a suite of unit tests for the GBC Travel Application's all critical components to ensure reliability and functionality.

Details:

- **Model Testing**: Write tests for your data models to validate business logic and data annotations. For example, ensure that a hotel's name is not empty and that pricing logic (if any) holds correctly.

- **Controller Testing**: Implement tests for your controllers to ensure they return the correct views or data and handle inputs as expected. Test different scenarios, including valid and invalid inputs.

- **Service Layer Testing**: If your application includes services (e.g., booking services), create unit tests to verify that these services behave as expected under various conditions.

- Please note, a video demonstration walkthrough of your unit tests in action, demonstrating successful execution and how they've contributed to the development process is compulsory. This portion of the demo **only**, can be conducted on your local development workspace.

AJAX Feature Implementation

Objective:
Enhance the interactivity and user experience of the GBC Travel Application by implementing AJAX. Specifically, you will use AJAX to dynamically load and display information within the application without the need for full page reloads.

Feature Details:

AJAX in Search Results:
- Implement AJAX in the travel services search functionality. When users search for flights, hotels, or car rentals, the search results should update dynamically without reloading the entire page.
- Provide visual feedback, such as a loader or spinner, while the search results are being fetched.

AJAX Form Submission:
- Implement AJAX for at least one form submission within the application, such as the booking confirmation or user review submission forms. Upon submission, the relevant section of the page should update dynamically to reflect the new data.

Implementation Instructions:
- Use jQuery or vanilla JavaScript to implement AJAX requests within your ASP.NET Core MVC application.
- Update your server-side code to handle AJAX requests and return partial views or JSON data as appropriate.
- Ensure that your AJAX implementation degrades gracefully; the application should still be functional without JavaScript.

Submission Requirements:
In your demonstration video, discuss and highlight the parts of the application that utilize AJAX, clearly showing the enhanced user experience compared to traditional full-page reloads.

Deployment Requirement: Microsoft Azure

Objective:
As part of Assignment 2, it is mandatory for your group to deploy the enhanced GBC Travel Application to Microsoft Azure. This will demonstrate your ability to take a web application from development to a live environment, an essential skill for any web developer.

Deployment Instructions:

1. **Azure Setup**:
   - If you have not already, set up an Azure account. You can utilize Azure for Students or any available Azure subscription.
   - Create a new Web App resource within the Azure portal for your travel application.

2. **Application Preparation**:
   - Ensure your application is production-ready. This includes removing any development settings, implementing proper error handling, and ensuring all features function as expected.
   - Update any necessary configurations to suit the production environment, such as database connection strings or API keys, ensuring they are securely stored using Azure's application settings.

3. **Deployment**:
   - Deploy your application to Azure. You can use GitHub Actions, Azure DevOps, or any other CI/CD pipeline, or manually deploy through the Azure portal or Visual Studio.
   - Once deployed, test your application thoroughly to ensure that all features work as expected in the live environment.

4. **Domain Name**:
   - While not required, you are encouraged to configure a custom domain name for your Azure web app for a more professional presentation.

5. **Submission Requirements**:
   - **Azure URL**: Your submission must include the URL to your deployed application on Azure. This URL should lead directly to the homepage of your Travel Application.

6. **Demonstration Requirements**:
   Your demonstration video for Assignment 2 must showcase the application running on the Azure environment, not on your local machine. Ensure that functionalities, particularly those newly added or enhanced in Assignment 2, are demonstrated using the live (azure) deployed version.

   Additional Notes:
   - Consider this deployment as a real-world exercise. The live environment should reflect a professional level of quality, security, and functionality.
   - Remember, deploying to Azure not only fulfills the assignment requirement but also adds value to your portfolio, demonstrating your ability to deliver complete, live web solutions.

Final Word on Implementation for Assignment 2

As you embark on enhancing your GBC Travel Application, it's important to note that this assignment's documentation does not include specific wireframes or design templates. This intentional omission aims to empower your creativity and originality. You are encouraged to design each page and feature according to your group's vision while maintaining a user-friendly and aesthetically pleasing interface.

This assignment provides comprehensive guidelines sufficient for completing the enhancements and integrating new functionalities. However, the exact implementation of these requirements involves your judgment and creativity. Approach each task thoughtfully, crafting code that best aligns with the objectives and improves the application.

Real-world software development often involves delving into uncharted territories, requiring research and proactive problem-solving. This assignment simulates such scenarios. You may encounter tasks that necessitate a deeper understanding of topics like authentication, data management, or new libraries and frameworks. Embrace these challenges as learning opportunities and demonstrate your team's ability to navigate and resolve complex issues.

Remember, the aesthetic and navigational aspects of the application are within your team's control. Your decisions on design and user interface should prioritize usability and innovation, factors that will be considered in your project's evaluation.

Feel free to expand beyond the basic requirements. If there are additional features or pages you believe would enhance your application, include them. Originality and thoughtful implementation are highly valued and can distinguish your work.

Finally, keep in mind that your project should embody the principles of the MVC architecture, a core focus of this course. Utilize models, views, and controllers effectively to build a well-structured and maintainable application.

By adhering to these guidelines, you not only fulfill the assignment's criteria but also advance your skills in developing sophisticated, real-world web applications.

Good luck, and I look forward to seeing your inventive solutions and the progress your group makes in enhancing the GBC Travel Application.

Assignment Submission Guidelines:

1. Video Requirement
   a. Create a Short Video presentation. Your presentation should start with an introduction, where it must display a PowerPoint (or Google Presentation), that is 1 (single) slide. The slide introduces each member of your group, again, at the very start of your video.

   b. The first (and only) slide of your presentation must include current images of you and your partner(s) (no avatars allowed) that are displayed appropriately. You must also include your Full Names, Student IDs, the Course Code, Course Name, Course Section and your Assignment information.

   c. Within the recording, you or your partner(s) will take turns demonstrating your program's functionality. You must show your site working properly. You will also construct an assignment status report, a single page checklist/report. Use the report during the video, to facilitate communication confirming where requirements where successfully implemented and/or where requirements failed to be implemented and why.

   d. You and your partner(s) will each share the responsibility in describing the code in your solution that drives the functionality of your program – you will want to do this part well and be very clear. Be intelligent/selective on what code segments you describe, I do not need to know how every line of code works.

   e. Sound for your video must at an appropriate level so that your voices may be clearly heard, and your screen resolution should be set so that your program's code and console details are clearly visible. In short, QA your videos. If your video is poor, assignment failures can/will be assigned.

   f. Your video should run no more than **<=15** minutes. If you exceed this time, I simply will **not** be able to watch them... resulting in a grade of **zero**.

2. The 1 team lead, must submit the following **3** components to **Brightspace** on behalf of the entire group:
   a. The 1-page status report – **mandatory**
   b. The (zipped) project source code – **mandatory**
   c. The group video file - **mandatory**
      i. You may find Vento useful to create this

   Steps to compress your project source - ONLY if you deem it too large
   d. Close your project and exit Visual Studio
   e. Navigate to your project folder, and make a copy (recommended)
   f. Remove can safely delete the following folders from your (copied) project directory:
      i. .vs
      ii. bin
      iii. obj
      iv. packages
   g. Now compress (**.zip**) the copied project folder and name accordingly

3. Be cautious **DO NOT** share your application with others. Complete failures will be assigned if code is shared. All assignments will be reviewed and analyzed strictly within these regards.

4. Late assignments are assigned a penalty of -10% per day (max 3 days late).

Good Luck!