COMP 2139 – Assignment 1

Due Date: Saturday, February 24th, 2024 (11:59 pm)
Team Size: Maximum of **3 students** per group.
Project Name: **GBC_Travel-Group**-<*Number*>

Objective:
The primary goal of this project is to develop a comprehensive, interactive web application that mirrors the functionality of renowned travel booking platforms like Expedia and Travelocity. Students will be given the opportunity to build an application that not only allows users to search for various travel-related services, including flights, hotels, and car rentals, but also incorporates a range of user-centric features. These features include user authentication, personalized booking management, real-time search with filtering and sorting capabilities, and an interactive user interface for seamless navigation and operation.

This project aims to challenge the students in groups, to integrate multiple aspects of web application development, including front-end and back-end development, database management, API integration, and responsive design principles. The application should demonstrate an understanding of ASP.NET Core MVC architecture, emphasizing the Model-View-Controller paradigm, and showcasing the students' ability to create a robust, scalable, and user-friendly web application.

Furthermore, students should be aware that this is the first of two interlinked assignments. The skills and the foundational application developed in this assignment will be crucial for the subsequent assignment (Assignment 2), where the focus will shift to integrating more advanced features and enhancements. Some of the complex functionalities and sophisticated features are intentionally deferred to Assignment 2, allowing students to build upon their initial work and progressively enhance their application.

By the end of this project, students will have gained hands-on experience in developing a feature-rich web application that addresses real-world requirements and challenges in online travel booking. The project is designed not only to enhance technical programming skills but also to foster critical thinking, problem-solving, and collaborative work, preparing students for future endeavors in the field of web development.

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 <u>**minimum**</u> project requirements have been summarized below.

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

Project Requirements (Compulsory Components)

The focus for this first Assignment is on building the user-facing parts of the application (search and booking functionality, guest user booking, listing of flights, hotels, and rentals).

Implement basic data entry for these services, possibly using seed data or simple forms, without extensive content management features (most deferred for Assignment 2).

Booking Functionality:

Requirement: Develop a booking system for the travel services (flights, hotels, car rentals).

Details:

- Implement a booking interface where users can select and confirm their bookings.
- Provide a summary page or confirmation screen with all booking details.
- Allow guest users (non-registered) to make bookings.

Highlights

- Booking Process: Users should be able to book flights, hotels, and car rentals.
- Booking Details: Include necessary details like dates, pricing, and confirmation.
- Guest Booking: Allow guest users (non-registered/non-logged-in) to make bookings, with the option to save their details for future use in Assignment 2.

Travel Search Functionality:

Requirement: Implement a search feature for flights, hotels, and car rentals.

Details:

- Allow users to search based on parameters like location, dates, number of passengers/guests, etc.
- Display search results with relevant details (e.g., flight times, hotel ratings, rental car models).
- Include sorting and filtering capabilities in the search results.

Highlights:

- Search Feature: Implement a mechanism to search for flights, hotels, and car rentals.
- Search Filters: Include intelligent filters such as location, date, price range, etc.
- Result Display: Search results should provide detailed information, with the option to view more details.

Guest User Booking:

Requirement: Create listings for flights, hotels, and car rentals.

Details:

- Use seed data or simple forms for basic data entry of service details.
- For each service type (flight, hotel, rental car), display essential information such as prices, availability, and specifications.
- Ensure that listings are easily accessible and navigable from the search results.

Accommodation and Transport Listing:

Requirement: Create listings for flights, hotels, and car rentals.

Details:

- Use seed data or simple forms for basic data entry of service details.
- For each service type (flight, hotel, rental car), display essential information such as prices, availability, and specifications.
- Ensure that listings are easily accessible and navigable from the search results.

Highlights

- Listing Feature: Allow clients to list accommodations and vehicles.
- Listing Details: Necessary details include name, description, pricing, availability dates, and images.
- Management Options: Clients should be able to review, delete, or update their listings.

Basic Data Entry Services:

Requirement: Implement basic data entry mechanisms for adding new flights, hotels, and car rental options.

Details:

- Create simple forms or interfaces for entering service details.
- Ensure data entry is functional and integrates with the search and booking features.

Application UX (User Experience) and Design:

Requirement: Develop a user-friendly interface with clear navigation.

Details:

- Design a visually appealing homepage with direct access to search and booking features.
- Include intuitive navigation aids like menus or search bars.
- Utilize HTML, CSS, and JavaScript to enhance the user experience.

Highlights

- Navigation: Implement a Navbar for ease of navigation.
- Footer: Include group information as previously described.
- Technologies: Encourage the use of HTML, CSS, and JavaScript for UI/UX enhancement.

Database Assignment Recommendations

For Assignment 1, the central component of your travel booking application will be the database, which is responsible for storing all the essential information required to power the platform. Given the nature of the application, your database should be designed to efficiently handle data related to travel services, including flights, hotels, and car rentals. Here are some key considerations:

1. **Incremental Development:** Start by building the database incrementally. Initially, focus on the core tables that are necessary for Assignment 1.

2. **Core Tables**: At this stage, your database should include tables for:
    - **Flights**: To store information about different flight options (e.g., airlines, departure and arrival times, prices).
    - **Hotels**: To manage details about various accommodation options (e.g., hotel name, location, pricing, amenities).
    - **Car Rentals**: To record details of car rental services (e.g., car models, rental companies, pricing, availability).

3. **Guest User Bookings**: Detailed user accounts and authentication are not necessary for Assignment 1, for now in Assignment 1, create a simplified structure for handling bookings made by guest users.

4. **Flexibility for Expansion**: Design your database schema with the foresight that it will be expanded in Assignment 2. This means considering how additional features, for example, such as user profiles, authentication, and itinerary management, could possibly integrate into your existing structure.

5. **Entity-Relationship Diagram (ERD) / Class Design**: Be meticulous in your ERD or class design. It should clearly represent the relationships between different data entities in your database. A well-thought-out ERD will simplify the process of expanding the database for Assignment 2.

6. **Data Integrity and Relationships**: Ensure that your database design maintains data integrity and accurately represents the relationships between different entities, such as the association between flights and bookings.

7. **Normalization**: Apply database normalization principles to avoid data redundancy and ensure data consistency.

8. **Utilization of Migrations**: The use of Entity Framework Core migrations is a compulsory component of this assignment. All migration files must be included in your project's version control repository.

Remember, the goal for Assignment 1 is to establish a solid foundation that not only meets the current requirements but also allows for seamless integration of more complex features in the next phase of the project. A well-designed database is crucial for the success of your application, both in its current state and as it evolves.

Final Word on Implementation

Emphasis on Creativity and Originality:

- **Design Freedom**: This assignment intentionally does not include wireframes or detailed design specifications. The design and layout of each page and the overall application are left to your group's creativity and discretion.
- **Expectation of Originality**: We encourage originality and creativity in your design choices. Remember, usability and aesthetics are subjective and will be part of the evaluation criteria.

Implementation and Problem-Solving:

- **Judgment and Implementation**: The provided requirements are detailed enough to guide you through the assignment, but they also leave room for your interpretation and decision-making in implementation.
- **Research and Exploration**: At time you may be expected to engage in research and exploration, especially for features not explicitly covered in the course.
- **Problem-Solving Skills**: Your approach to solving the challenges posed by the requirements will be a key aspect of evaluation. This is an opportunity to showcase how your group can tackle real-world problems.

Design Considerations:

- **Final Design Decisions**: All final design decisions, including the user interface, navigation, and aesthetics, are your group's responsibility.
- **MVC Focus**: Ensure that your design adheres to the MVC (Model-View-Controller) architecture, making appropriate use of models, views, and controllers. This is central to the course and critical for your project.

Flexibility and Expansion:

- **Additional Pages**: Feel free to add extra pages or features that you believe will enhance your application, as long as the core requirements are met.
- **Interpretation and Originality**: While the requirements document sets out the expected functionalities, your team has the freedom to interpret and expand upon these. Original and innovative implementations are welcome and will be considered positively in evaluations.

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 **~5-10** 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 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 OBS Studio useful to create this
   d. Public (Cloud) Deployment URL
   e. Private GitHub URL

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