Activity 1: Using Microsoft Copilot to Generate Blazor Code

Activity introduction: generative AI tools like Microsoft Copilot can streamline your workflow, especially when building the foundational components of a project. In this activity, you'll use Copilot to create essential elements for the EventEase app, an event management application you'll develop over a series of activities. This is the first of three activities for building the EventEase app. The work you complete here will form the foundation for debugging and optimization tasks in Activity 2 and expansion in Activity 3.

Step 1: Review the scenario

To begin, review the following scenario to understand the intent of this project.

EventEase is a fictional company specializing in corporate and social event management. They've tasked you with creating a web application to allow users to:

- Browse events with details such as event name, date, and location.
- Navigate seamlessly between pages like event details and registration.

As part of this development team, your responsibility is to build and refine the front-end using Blazor while leveraging Microsoft Copilot to accelerate the coding process.

Step 2: Review the task requirements

Your first task is to create the Event Card component and set up basic routing for the EventEase app. These features will act as the foundation for the application and allow future debugging and expansion.

Step 3: Generate a basic Blazor component with Copilot

Open your Visual Studio sandbox environment and ensure Copilot is enabled. Then, create a new Event Card component.

- Use Copilot's suggestions to define the structure of the component.
- Include fields for event name, date, and location.

Step 4: Implement two-way data binding with Copilot

Add data binding to dynamically display event details.

- Use mock data or a simple data model.
- Use Copilot to suggest and refine binding syntax.

Step 5: Set up routing between pages

Finally, you’ll work on the routing between pages.

- Create navigation links between the event list, details, and registration pages.
- Use Copilot to generate and verify routing paths, ensuring best practices.

Step 6: Save your work

By the end of this activity, you will have:

- A working Event Card component with fields and two-way data binding.
- Routing functionality between pages of the EventEase app.
- A foundational codebase, ready for debugging in the next activity.

Ensure your code is saved in your sandbox environment. The components and functionality you’ve built here will be used in Activity 2 for debugging and optimization.

Activity 2: Using Microsoft Copilot for Debugging and Optimization

Activity Introduction: debugging is an essential skill in development, and Microsoft Copilot can simplify the process. In this activity, you'll debug and optimize the code you created in Activity 1 for the EventEase app, ensuring reliability and performance. This is the second of three activities. The debugged and optimized code will serve as the base for adding advanced features in Activity 3.

Step 1: Review the recap

In Activity 1, you created a basic Event Card component with data binding and routing functionality for the EventEase app. However, initial testing revealed some issues:

- Data binding fails with invalid inputs.
- Routing generates errors when navigating to non-existent pages.
- The performance of the event list is slow with large data sets.

Your task is to resolve these issues and optimize the app using Microsoft Copilot.

Step 2: Review the identified bugs

Open your sandbox environment to access the code created in Activity 1. Use Copilot to analyze and identify bugs, such as:

- Missing input validation in the Event Card.
- Errors in routing for invalid paths.
- Performance bottlenecks in the event list.

Step 3: Debug with Copilot

Apply Copilot's suggestions to fix identified issues:

- Add validation to ensure only valid data is processed in the Event Card.
- Update routing logic to gracefully handle invalid paths.
- Optimize performance by improving how event data is rendered.

Step 4: Test and validate fixes

Test the updated code:

- Verify data binding works for all edge cases (e.g., invalid or empty data).

- Test routing to confirm errors are handled correctly.
- Measure performance improvements for larger event datasets.

Step 5: Save your work

By the end of this activity, you will have:

- Debugged the Event Card and routing functionality.
- Optimized the app’s performance and reliability.
- A clean and efficient codebase ready for advanced features in Activity 3.

Save your debugged and optimized code in your sandbox environment. This refined version will serve as the foundation for the comprehensive project in Activity 3.

Activity 3: Develop a Comprehensive Blazor Project with Microsoft Copilot

Activity Introduction: in this final activity, you'll bring together everything you've learned to complete the EventEase app. Using Microsoft Copilot, you'll expand the app with advanced features, test it thoroughly, and prepare it for deployment. This activity builds on the debugged and optimized code from Activity 2.

Step 1: Review the recap

In the previous activities, you:

- Generated foundational components for the EventEase app (Activity 1).
- Debugged and optimized the code for performance and reliability (Activity 2).

Now, you'll enhance the app with advanced features, including state management, form validation, and a user-friendly interface.

Step 2: Plan the app's final structure

Review your code from Activities 1 and 2. Use Copilot to outline the remaining features, such as:

- A Registration Form with validation.
- A User Session Tracker for state management.

Step 3: Add advanced features with Copilot

Use Copilot to implement:

- A Registration Form with data validation for user input (e.g., name, email).
- State management to track user sessions.
- An Attendance Tracker to monitor event participation.

Step 4: Test and validate the final app

Test all features:

- Verify the Registration Form handles errors gracefully.
- Confirm state management ensures user session continuity.
- Check overall app performance for responsiveness and reliability.

Step 5: Prepare the app for deployment

Use Copilot to optimize the code for production readiness:

- Ensure the code follows best practices.
- Remove unnecessary dependencies.

Step 6: Save your work

By the end of this activity, you will have:

- A fully functional and deployable EventEase app.
- Demonstrated proficiency in using Microsoft Copilot for Blazor development.

Save the completed EventEase app in your sandbox environment. Prepare a brief summary of how Copilot assisted in each step of the development process.