**Supporting Explanation (3.2)**

**1. Structure and Approach**

**HTML Structure**

I structured my HTML with semantic elements to enhance accessibility, maintainability, and search engine optimization (SEO). For example:

* Used <header> for navigation and branding, <main> for content, and <footer> for additional links or copyright information.
* Semantic tags like <section>, <article>, and <nav> were used to create a logical hierarchy, making the page easier to navigate for assistive technologies.
* Ensured the forms used <label> elements correctly associated with form controls using for attributes for accessibility.

**CSS and Responsive Design**

* Implemented a mobile-first design approach to ensure the site works seamlessly on smaller devices and scales up gracefully to larger screens.
* Used media queries for breakpoints at common screen widths (e.g., 768px for tablets and 1024px for desktops).
* Flexbox and Grid layouts were used for layout responsiveness and alignment consistency.

**Routing and Structure**

* Adopted the MVC (Model-View-Controller) pattern to separate concerns, making the application modular and easier to maintain.
* Routes were defined in index.php, and logic was passed to appropriate controllers, ensuring clarity in request handling.
* Views were kept lightweight with no direct database interaction, ensuring code reuse and maintainability.

**2. Verification and Validation**

**Input Verification and Validation**

* Server-side validation was implemented in the BookingController to ensure data integrity. For example, checking the party size range (1–6) and ensuring the selected time is within service timings.
* Client-side validation, including input type constraints and JavaScript, provides immediate feedback to users.
* Error messages were user-friendly, providing actionable guidance to correct inputs.

**Security Measures**

* Sanitized all user inputs using PHP functions like htmlspecialchars and filter\_var to prevent XSS attacks.
* Prepared SQL statements with parameterized queries using PDO to prevent SQL injection.
* Added session-based authentication for secure login and restricted access to admin functionalities.

**3. Accessibility and Responsiveness**

**Accessibility**

* Used ARIA roles and landmarks where appropriate, such as role="alert" for error messages.
* Ensured sufficient colour contrast between text and background for readability.
* Added focus styles for interactive elements like buttons and links to improve keyboard navigation.

**Responsiveness**

* Designed forms, tables, and navigation to adapt seamlessly to various screen sizes.
* Used percentage-based widths and rem units for font sizes to ensure scalability across devices.

**4. Key Decisions and Trade-Offs**

* **Email Confirmation:** Constructed .eml files instead of sending actual emails due to server limitations, ensuring compliance with requirements.
* **Calendar Integration:** Implemented .ics file generation for calendar events, providing an easy way for users to add bookings to their calendars.
* **Security vs. User Experience:** Balanced robust input validation with user-friendly error messages to enhance usability without compromising security.