**Project Plan and Folder Structure Documentation**

**1. Overview**

The project folder contains assets, scripts, styles, and HTML files necessary to create a web application. Below is a breakdown of the folder structure, the purpose of each directory, and the plan for managing the project.

**2. Folder Structure and Purpose**

**Root Folder: /02-01-25**

* **Files**:
  + index.html: The homepage of the website.
  + offers.html: Displays special offers or promotions.
  + payment.html: Manages payment-related features.
* **Subfolders**:
  + images: Stores all image assets.
  + pages: Contains additional HTML pages.
  + scripts: Includes JavaScript files for functionality.
  + styles: Holds CSS files for styling.

**Subfolder: images**

* **Purpose**: Contains all image assets used across the website.
* **Files**:
  + image1.PNG, image2.PNG, image3.jpeg, image3.PNG, image4.jpeg: Generic image assets for website elements.
  + logo.jpg: The website’s logo.
  + oldhtm.txt: A legacy file that might contain notes or backup HTML content.

**Subfolder: pages**

* **Purpose**: Additional web pages for specific content.
* **Files**:
  + customer.html: Page related to customer information or services.
  + staff.html: Page for staff information or management.

**Subfolder: scripts**

* **Purpose**: JavaScript files to handle dynamic and interactive features.
* **Files**:
  + auth.js: Handles user authentication.
  + config.js: Configuration settings for the application.
  + customer311224.js: Customer-specific functionality.
  + customeriddrop.js: Dropdown-related operations for customer IDs.
  + google-sheets-api.js: Integration with Google Sheets for data handling.
  + logout.js: Handles user logout functionality.
  + order.js: Manages order-related processes.
  + pagination.js: Adds pagination functionality to tables or lists.
  + payment.js: Payment-related features.
  + staff1.js: Staff-specific operations.
  + table.js: Handles dynamic table generation and interactions.

**Subfolder: styles**

* **Purpose**: CSS files for styling the website.
* **Files**:
  + offers.css: Styles specific to the offers page.
  + style.css, styles.css: General styling files for the website.
  + table.css: Styles for tables.

**3. Project Plan**

**Phase 1: Initial Setup and Review**

1. **Organize Assets**: Verify all files are correctly placed in their respective directories.
2. **File Cleanup**: Remove unused or redundant files (e.g., oldhtm.txt if unnecessary).
3. **Code Audit**: Review all scripts and styles for optimization and security.

**Phase 2: Development**

1. **Homepage Setup**: Ensure index.html is fully functional with linked styles and scripts.
2. **Page Integration**:
   * Link customer.html and staff.html in the navigation.
   * Validate the functionality of offers.html and payment.html.
3. **Feature Implementation**:
   * Test all JavaScript files, ensuring features like authentication, order handling, and pagination work seamlessly.
   * Integrate Google Sheets API if necessary.

**Phase 3: Testing and Deployment**

1. **Cross-Browser Testing**: Ensure compatibility with major browsers.
2. **Responsive Design**: Validate that the site is mobile-friendly.
3. **Deployment**: Publish the site to a web server.

**Phase 4: Maintenance**

1. Regular updates to content and styles.
2. Performance monitoring and bug fixes.
3. Integration of new features based on user feedback.

**4. Notes**

* **Dependencies**: Ensure all external libraries or frameworks used are up-to-date.
* **Documentation**: Add comments in the codebase for future developers.
* **Backup**: Create periodic backups of the project to prevent data loss.