DIGITAL MEAL PLAN | CORE | Composer and MVC Templating

Overview

Your company has been contracted to create a comprehensive digital meal plan for Catering4Schools GmbH. The system will show menus, allow pre-orders, support filters for allergens and dietary preferences, and handle payments. The project requires professional PHP development practices with Composer for dependency management and a custom templating system to separate concerns between backend logic and frontend presentation.

Definition of Done

The following quality standards apply to ALL user stories: - Code follows PSR-4 autoloading standards and professional PHP practices - All functionality is tested and working correctly - User interface is responsive and accessible across devices - Code is clean, documented, and maintainable - Security best practices are implemented (input validation, sanitization) - All deliverables meet professional development standards

User Story 1: Composer Setup

As a Developer I want to set up composer, so that I have a package manager available for professional development.

Acceptance Criteria

- Composer is available on the local development environment
- Composer can be called via the command line
- Composer is updated to the latest version

User Story 2: Project Initialization

As a Developer I want to initiate a new composer project in my working directory, so that the rest of my team can replicate my setup.

Acceptance Criteria

- A composer.json exists
- A composer.lock exists
- Autoloading according to PSR-4 is enabled for the project

User Story 3: QR Code Package Integration

As a Developer I want to require the package "endroid/qr-code" in my project, so that I can generate QR codes for meal plan access.

Acceptance Criteria

- endroid/qr-code is listed in the required section of the composer.json
- endroid/qr-code is downloaded on the local development environment
- Autoloading for endroid/qr-code is tested and working

User Story 4: Meal Plan Prototype

As a Developer I want to create a responsive prototype HTML page, containing demonstration content about the meal plans.

Acceptance Criteria

- A plain HTML prototype exists
- There is a short information block about at least 3 different meal options
- The prototype is responsive and mobile-friendly

User Story 5: Basic Templating Engine

As a Developer I want to create a basic templating engine based on the PHP functions "fread" and "str_replace", using HEREDOC for longer text in variables, in order to decouple the HTML view from the code.

Acceptance Criteria

- The plain HTML prototype has placeholders which are replaced programmatically
- The plain HTML prototype can be edited/enhanced with the help of placeholders without the need of changing the PHP Code
- Meal data is stored in PHP arrays and rendered through the templating system

User Story 6: Dynamic Content Rendering

As a Developer I want to create a functionality in my custom templating engine that supports loops, so that meal options which are added in the code are automatically rendered without having to alter the template.

Acceptance Criteria

- The plain HTML prototype can display an arbitrary number of meals based on the number of meals created in the code
- Meal information is dynamically populated from PHP data structures
- The templating system supports iteration over meal arrays

User Story 7: QR Code Generation Integration

As a Developer I want to use the package endroid/qr-code to generate QR codes for meal plan access and display them on the meal pages.

Acceptance Criteria

- QR codes for meal plan access are generated and displayed
- Each meal page shows a QR code containing the access information
- The QR codes are properly integrated into the templating system
- The correct readability of the QR Codes has been tested with a smartphone

User Story 8: Interactive QR Code Generator

As a Developer I want to create a styled webpage with a form where users can input a meal ID and generate a QR code, so that they can create access QR codes for any meal.

Acceptance Criteria

- A responsive website with basic styling exists
- A form exists where users can input any meal ID
- After submitting the form, a QR code for the entered meal ID gets displayed on the page
- The form integrates seamlessly with the existing templating system
- Error handling is implemented for invalid meal ID formats