Online Marketplace - Project Overview

Leonid Malakhov

Date: 09.01.2024
Subject: KIV/WEB
Application name: Online Marketplace

Technologies Used

Frontend:

- HTML
- CSS
- JavaScript

Backend:

- PHP

Database:

- MySQL

Application Directory Structure

```
application/
        controllers/
                // list of controllers
        core/
                // core php files
        css/
                // css style
        includes/
                // navbar
        models/
                // list of models
        uploads/
                // folder for uploaded files
        views/
                // list of views
```

Application Architecture

MVC Architecture:

Model:

- Manages data logic, database interaction, and overall application logic.
- Ensures secure data manipulation and interaction with the database.

View:

- Presents data to the user.
- Defines HTML structure, templates for product lists, order details, forms, etc.

Controller:

- Acts as an intermediary between Model and View.
- Identifies controller and action based on the URL.
- Triggers methods in the Model for data retrieval.
- Passes data to the View and invokes its rendering method.

EntryPoint: index.php - initializes and routes requests.

Implementation in the Context of an Online Marketplace

Model:

- Classes for user accounts, products, orders, etc.
- Ensures secure data manipulation and interaction with the database.

View:

- Defines HTML structure, templates for displaying data.
- Handles appearance and interactivity.

Controller:

- Identifies controller and action based on the URL.
- Triggers methods in the Model for data retrieval.
- Passes data to the View and invokes its rendering method.

Users Logins and Passwords:

- Buyer: cashtad@gmail.com - 123
- Seller: seller@test.com - 123
- Admin: admin@gmail.com - 123

Requirements for Running a Web Server

Compatible web server (Apache, Nginx)
PHP 7.0 or later with extensions (mysqli, pdo, openssl)
Supported relational database (MySQL, MariaDB)
Modern web browser

Deployment Instructions

Clone the repository
Set up the database
Configure connection settings in config.php
Adjust the web server to point to the application's root
Access the application through the server's URL (e.g., http://localhost/*YOUR_PAGE*)
Ensure proper permissions and test by opening the configured URL in a browser.

Conclusion

In conclusion, the application architecture demonstrates a well-structured MVC pattern, providing a clear separation of responsibilities. Each class and module serve specific purposes, facilitating code maintenance and future development. Controllers act as a bridge between the user interface and data model, while models handle data manipulation and storage. This architecture lays a solid foundation for building robust and scalable web applications, showcasing features of the modern MVC design pattern, enhancing testability, and ensuring a clear separation of concerns.