UNIVERSIDADE DO CENTRO OESTE DE SANTA CATARINA – CAAMPUS VIDEIRA

ALUNOS: JOÃO VITOR BORGES GODINHO, JULIO ORSO, JULIANNA ORSO

PROFESSOR: Leandro Otavio Cordova Vieira.

DISCIPLINA: PROGRAMAÇÃO III

RELATÓRIO PROJETO FINAL, PARTE 2 – FRAMEWORKS PHP (LAMINAS FRAMEWORK)

1. Introdução

O projeto consiste na criação de um sistema de gerenciamento de estacionamento utilizando o Laminas (anteriormente Zend Framework). O objetivo principal é oferecer uma plataforma robusta e segura para reserva de vagas de estacionamento, tanto para clientes quanto para administradores.

2. Arquitetura

Arquitetura MVC

A arquitetura MVC (Model-View-Controller) é um padrão de arquitetura de software amplamente utilizado para desenvolver interfaces de usuário em aplicações web. Ela separa a aplicação em três componentes principais, cada um com responsabilidades específicas:

1. Model (Modelo):
    - Representa a camada de dados da aplicação.
    - Gerencia o acesso aos dados, regras de negócio e lógica da aplicação.
    - Normalmente interage com o banco de dados para recuperar e armazenar informações.
2. View (Visão):
    - Responsável pela apresentação dos dados ao usuário.
    - Recebe informações do Modelo e as apresenta de maneira formatada.
    - Geralmente é uma interface gráfica que os usuários interagem.
3. Controller (Controlador):
    - Atua como intermediário entre o Modelo e a Visão.
    - Responde às solicitações do usuário e invoca as operações apropriadas no Modelo.
    - Atualiza a Visão com os dados resultantes para exibição ao usuário.

Funcionamento:

- Quando o usuário interage com a aplicação (como clicar em um link ou enviar um formulário), a requisição é inicialmente enviada ao Controlador.

- O Controlador processa a requisição, interage com o Modelo conforme necessário (recupera, atualiza ou deleta dados) e decide qual View deve ser renderizada com base nas ações do usuário e nos dados disponíveis.
- A View recebe os dados do Controlador e os apresenta ao usuário de forma visualmente compreensível.

3. Funcionalidades Implementadas

Registro e Login

Registro de Usuários

Os usuários podem se registrar no sistema fornecendo informações como nome de usuário, senha (criptografada utilizando password_hash), email e plano de preferência (pré-pago ou pós-pago).

Login

Mecanismo de login seguro, verificando as credenciais do usuário no banco de dados e mantendo a sessão ativa utilizando Laminas\Session\Container.

Reserva de Vagas

Disponibilidade de Vagas

Os clientes podem visualizar vagas disponíveis e realizar reservas selecionando o horário de check-in e check-out.

Cálculo de Tarifas

As tarifas são calculadas com base no plano escolhido pelo cliente (pré-pago ou pós-pago) e no tempo de ocupação da vaga.

Painel do Usuário

Atualização de Informações Pessoais

Os usuários podem atualizar suas informações pessoais, como nome de usuário e senha.

Histórico de Reservas

Histórico detalhado das reservas anteriores, incluindo datas, vagas reservadas e valores pagos.

Painel de Administração

Gerenciamento de Usuários

Funcionalidades para administradores gerenciarem usuários, incluindo adição, edição e exclusão de contas de usuário.

Gerenciamento de Vagas de Estacionamento

Administração completa das vagas de estacionamento, permitindo adição, edição e exclusão de vagas conforme necessário.

Relatórios Gerenciais

Reservas de Vagas de Estacionamento

Geração de relatórios detalhados sobre as reservas de vagas de estacionamento, mostrando dados como número de vagas reservadas, períodos mais movimentados, etc.

Vendas e Receitas

Relatórios financeiros detalhando as receitas geradas pelo sistema, categorizadas por tipo de plano (pré-pago, pós-pago), proporcionando uma visão clara das fontes de receita do sistema.

4. Desafios e Soluções

1. Arquitetura e Implementação MVC:

- Desafio: Compreensão e implementação correta da arquitetura MVC para garantir a separação clara de responsabilidades entre modelos, visões e controladores.
- Solução: Utilização dos recursos e convenções fornecidos pelo Laminas (Zend Framework) para estruturar o projeto de forma modular e escalável. Implementação de controladores robustos e modelos bem definidos para facilitar a manutenção e expansão do sistema.

2. Segurança e Controle de Acesso:

- Desafio: Implementação de medidas eficazes de segurança para proteger os dados sensíveis dos usuários e do sistema.
- Solução: Utilização de autenticação baseada em sessão com Laminas\Session\Container para gerenciar o acesso dos usuários. Implementação de hash seguro de senhas usando password_hash() e password_verify() do PHP para armazenamento seguro no banco de dados. Controle de acesso baseado em níveis de permissão para garantir que apenas administradores tenham acesso a funcionalidades sensíveis.

3. Geração de Relatórios Gerenciais:

- Desafio: Desenvolvimento de um sistema de controle
- Solução: Utilização de consultas SQL complexas para extrair dados relevantes do banco de dados MySQL. Implementação de páginas específicas no painel de administração para apresentação.

5. Conclusão

Realização do Projeto

Apesar da dificuldade inicial com os componentes e estrutura do Laminas Framework, com o passar do tempo, se acostumando com o padrão do código, foi se tornando cada vez mais prática trabalhar com o framework. A implementação dos módulos gera um bom adianto no trabalho, tornando-o mais fácil.

Melhorias Futuras

Fazer um front-end mais agradável. Implementar um sistema de pagamento. Melhorar as condições de segurança com autenticações. Incrementar ainda mais as funcionalidades já existentes para melhor servir os usuários. Exportação de relatórios em formatos como PDF