Review Feature Software Documentation

Overview

This document provides an overview of the review feature for the GoFly platform. It covers the components and functionality required to allow customers to create, edit, and view reviews of their experiences. The feature aims to help users make informed decisions by providing them with valuable feedback from other customers.

Components

- reviews_list.css - A CSS file containing the styling for the review components.
- edit_review.php - PHP script responsible for handling the review editing process.
- reviews_list.php - PHP script for displaying the list of reviews.
- reviews.php - PHP script for handling the creation and submission of new reviews.
- submit_review.php - PHP script responsible for processing submitted reviews.

Tools and Technologies

The following tools and technologies have been used to implement the review feature:

- PHP: A server-side scripting language used for web development. PHP has been used to create the scripts responsible for handling the creation, editing, and displaying of reviews.

- HTML/CSS: The standard markup and styling language for creating web pages. HTML and CSS have been used to structure and style the review forms and the review display.
- MySQL: A relational database management system for storing and managing the submitted reviews. MySQL has been used as the backend storage for the review data.
- JavaScript/jQuery: A programming language and library used for creating interactive web pages. JavaScript and jQuery have been used to enhance the user experience when interacting with the review forms and navigating through the reviews.

Workflow

- A user logs into their GoFly account.
- The user navigates to the "Reviews" page by clicking the "Reviews" option on the top navigation bar.
- The user can either create a new review or edit an existing one.
- To create a new review, the user fills out the review form and clicks the "Submit" button.
- To edit an existing review, the user clicks the 3-dotted bar next to their review and selects the "Edit" option. This will redirect them to the "Edit Your Review" page. The user modifies their review and clicks the "Update" button to save the changes.
- After submitting or updating a review, the user is redirected to the "Our Customer Reviews" page, where they can view all the submitted reviews.

Implementation Details

- Creating and Submitting Reviews

- The reviews.php script generates the review form with the required fields: full name, comment box, and 1 to 5-star rating selection.
- The "Submit" button triggers the submit_review.php script, which processes the submitted review, saves it in the MySQL database, and displays a confirmation message to the user.
- The reviews_list.php script retrieves the submitted reviews from the MySQL database and displays them on the "Our Customer Reviews" page.
- Editing and Updating Reviews
- The edit_review.php script generates the edit review form with the same fields as the create review form, pre-filled with the user's existing review data retrieved from the MySQL database.
- The "Update" button triggers an update process, which saves the modified review to the MySQL database and displays a confirmation message to the user.
- The reviews_list.php script updates the displayed reviews on the "Our Customer Reviews" page to reflect the changes made by the user.

Future Enhancements

- Add the ability for users to delete their reviews.
- Implement pagination for displaying large numbers of reviews.
- Allow users to sort and filter reviews based on criteria such as date, rating, or helpfulness.

Delete Account Feature Software Documentation

Overview

This document provides an overview of the delete account feature for the GoFly platform. It covers the components and functionality required to allow users to permanently delete their accounts and associated data from the system. The feature aims to give users control over their personal information and ensure data privacy.

Components

- deleteProfile.html - An HTML file containing the form for users to confirm their intention to delete their account.
- deleteProfile.php - PHP script responsible for handling the account deletion process.
- successDeleteProf.html - An HTML file displaying a confirmation message upon successful account deletion.
- login.css - A CSS file containing the styling for the delete account components.
- landing.css - A CSS file containing the styling for the deleted account components.

Tools and Technologies

The following tools and technologies have been used to implement the delete account feature:

- PHP: A server-side scripting language used for web development. PHP has been used to create the scripts responsible for handling the account deletion process.
- HTML/CSS: The standard markup language and styling language for creating web pages. HTML and CSS have been used to structure and style the delete account confirmation form and the success message.
- MySQL: A relational database management system for storing and managing user accounts. MySQL has been used as the backend storage for the user account data.

Workflow

- A user logs into their GoFly account.
- The user navigates to the "Delete Account" page.
- The user is prompted to enter their current password to confirm their intention to delete the account.
- Upon successful password verification, the deleteProfile.php script permanently deletes the user's account and associated data from the MySQL database.
- The user is redirected to the successDeleteProf.html page, where they are shown a confirmation message that their account has been deleted successfully.

Implementation Details

Account Deletion Confirmation

- The deleteProfile.html file generates a form that prompts the user to enter their current password to confirm their intention to delete their account.

- The "Submit" button triggers the deleteProfile.php script, which verifies the entered password against the stored password in the MySQL database.

Deleting Account and Associated Data

- Upon successful password verification, the deleteProfile.php script proceeds to permanently delete the user's account and associated data from the MySQL database.
- The script also destroys the user's session to log them out of the system.

Displaying Confirmation Message

- After the account deletion process is complete, the user is redirected to the successDeleteProf.html page.
- The successDeleteProf.html file displays a confirmation message to inform the user that their account has been deleted successfully.

Future Enhancements

- Add a two-step verification process (e.g., email or SMS confirmation) to further ensure the security of account deletion.
- Implement a grace period during which users can restore their deleted accounts, in case they change their mind or delete their accounts accidentally.
- Allow users to download a copy of their data before deleting their account, for backup purposes.

Sort Listing Feature Software Documentation

Overview

This document provides an overview of the Sort Listing feature for the Flight Booking platform. It covers the components and functionality required to allow users to sort flight listings based on various criteria, such as price, duration, airline, and destination. The feature aims to help users find the most suitable flight options according to their preferences.

Components

- displaylist.php - PHP script responsible for displaying flight listings and the sort selection form.
- sort_listing.php - PHP script responsible for sorting flight listings based on the user's chosen criteria.
- display.css - A CSS file containing the styling for the flight listings display.

Tools and Technologies

The following tools and technologies have been used to implement the Sort Listing feature:

- PHP: A server-side scripting language used for web development. PHP has been used to create the scripts responsible for handling the flight listing sorting process.
- HTML/CSS: The standard markup and styling language for creating web pages. HTML and CSS have been used to structure and style the flight listings display and the sort selection form.

- MySQL: A relational database management system for storing and managing flight listings data. MySQL has been used as the backend storage for flight listings information.
- JavaScript: A programming language used for creating interactive web pages. JavaScript has been used to enhance the user experience when interacting with the sort selection form and navigating through the site.
- Google Fonts: A web font service that provides custom fonts for web pages.

Workflow

1. A user logs into their Flight Booking account.
2. The user navigates to the flight listings page, where all available flight options are displayed.
3. The user selects a sorting criteria from the "Sort by" dropdown menu.
4. The flight listings are updated and sorted based on the chosen criteria.

Implementation Details

Displaying and Sorting Flight Listings

- The displaylist.php script generates the flight listings and the "Sort by" dropdown menu with the available sorting options: Price High-Low, Price Low-High, Duration, Airline, and Destination.
- The flight listings are retrieved from the MySQL database and displayed by default without any sorting applied.
- The user selects a sorting option, which triggers the sort_listing.php script through a form submission.

- The sort_listing.php script processes the chosen sorting option, fetches the sorted flight listings from the MySQL database, and stores them in a session variable.
- The user is redirected back to the displaylist.php page, where the sorted flight listings are displayed.

Future Enhancements

- Add advanced filtering options, allowing users to filter flight listings based on departure time, number of stops, and other relevant criteria.
- Implement a "Save search preferences" feature, allowing users to save their preferred sorting and filtering options for future use.
- Improve the user interface for the flight listings page by providing a more consistent and visually appealing design.
- Optimize the sorting and filtering process to enhance performance, especially when dealing with a large number of flight listings.
- Add support for additional flight information, such as layover duration and details, to provide users with a more comprehensive overview of the available flight options.

Weather Forecast Feature Documentation

Introduction:

This documentation provides an overview of the weather forecast feature implemented in the GoFly platform. The feature allows users to search for the current weather information of a city using OpenWeatherMap API. The weather forecast feature is accessible from the platform's navigation bar.

Files:

The feature is implemented in the following files:

- weather.php - Main PHP file containing the HTML structure and JavaScript code for fetching weather data.
- weather.css - CSS file containing the styling for the weather forecast feature.

Tools and Technologies:

The weather forecast feature is implemented using the following tools and technologies:

- HTML5 - Used for creating the structure of the weather forecast page.
- CSS3 - Used for styling the weather forecast page.
- JavaScript - Used for fetching weather data from OpenWeatherMap API and updating the DOM with the fetched data.
- PHP - Used for session management and handling user authentication.
- OpenWeatherMap API - A RESTful API used for fetching the current weather data of the searched city.

Implementation Details:

weather.php

- The weather.php file is responsible for rendering the weather forecast page. The page consists of a search bar for users to enter the city name, a submit button to fetch the weather data, and a container to display the weather information.

- Upon loading the page, the JavaScript code fetches the weather data for a default city ("New York") using the OpenWeatherMap API. The API key is stored in a constant variable named apiKey.
- When the user submits the search form with a city name, the JavaScript code fetches the weather data for the entered city and updates the DOM with the fetched data.

weather.css

- The weather.css file contains the styling for the weather forecast page. It includes styles for the search bar, submit button, weather information container, and responsive design for smaller screens.

Usage:

To access the weather forecast feature, users must be logged in. After logging in, they can click the "Weather" option in the navigation bar to open the weather forecast page. Users can then enter the city name in the search bar and click the "Search" button to fetch the weather information for the entered city.

Testing:

To test the weather forecast feature, users can follow these steps:

1. Log in to the GoFly platform using valid credentials.
2. Click the "Weather" option in the navigation bar.
3. Verify that the weather forecast page is displayed with a search bar and a default city's weather information.
4. Enter a valid city name in the search bar and click the "Search" button.
5. Verify that the weather information for the entered city is displayed.

6. Enter an invalid city name in the search bar and click the "Search" button.
7. Verify that an empty box is displayed.

Future Improvements:

1. Add a feature to display a 5-day weather forecast.
2. Implement an auto-suggest feature for the city search bar to improve user experience.
3. Add geolocation support to automatically display weather information for the user's current location.

Edit Profile Functionality Software Documentation

Overview

This document provides an overview of the Edit Profile functionality for the GoFly platform. It covers the components and functionality required to allow users to update their personal information, such as first name, last name, phone number, and password. The feature aims to help users keep their account information up-to-date.

Components

- editprofile.php - PHP script responsible for handling the profile updating process.
- profile.php - PHP script for displaying the user's profile information and the Edit Profile form.
- login.css - A CSS file containing the styling for the login form, reused for the Edit Profile form.
- landing.css - A CSS file containing the styling for the landing page, reused for the Edit Profile form's navigation bar.
- land.js - A JavaScript file containing the functionality for the responsive navigation bar.

Tools and Technologies

The following tools and technologies have been used to implement the Edit Profile functionality:

- PHP: A server-side scripting language used for web development. PHP has been used to create the scripts responsible for handling the profile information updating process.
- HTML/CSS: The standard markup and styling language for creating web pages. HTML and CSS have been used to structure and style the Edit Profile form and the user's profile display.
- MySQL: A relational database management system for storing and managing user data. MySQL has been used as the backend storage for user profile information.
- JavaScript: A programming language used for creating interactive web pages. JavaScript has been used to enhance the user experience when interacting with the Edit Profile form and navigating through the site.
- Google Fonts: A web font service that provides custom fonts for web pages.

Workflow

- A user logs into their GoFly account.
- The user navigates to their profile page by clicking the profile icon in the top right corner.
- The user clicks the "Edit Profile" button, which redirects them to the Edit Profile page.
- The user updates the desired fields (first name, last name, and phone number).
- The user clicks the "Save" button to submit the form.

- The updated information is displayed on the profile page after a successful submission.

Implementation Details

- Creating and Submitting Updates to Profile Information
- The profile.php script generates the Edit Profile form with the required fields: first name, last name, and phone number.
- The form fields are pre-populated with the user's current information retrieved from the MySQL database.
- The "Save" button triggers the editprofile.php script, which processes the submitted updates, saves them in the MySQL database, and displays a success message to the user.
- Editing and Updating the Password
- The profile.php script provides a link to the change_password.html page, where users can update their password.
- The password update process is handled separately from the rest of the profile information.

Future Enhancements

- Add the ability for users to update their email address and receive a confirmation email to ensure the new email is valid.
- Implement additional input validation on the client side to provide instant feedback on invalid inputs before submitting the form.
- Improve the user interface for the Edit Profile page by providing a more consistent and visually appealing design.
- Integrate the password update process into the Edit Profile form for a more streamlined user experience.

- Add support for profile pictures, allowing users to upload and update their avatars.

Display Hotel Listing Software Documentation

Overview

This document provides an overview of the hotel display feature for the GoFly platform. It covers the components and functionality required to allow administrators to view and manage the hotels listed on the platform. The feature aims to help administrators monitor the hotels' status and make informed decisions by providing them with relevant information.

Components

- admin_hotelDisplay.css - A CSS file containing the styling for the hotel display components.
- admin_hotelDisplay.php - A PHP script responsible for displaying the list of hotels and their details to the administrator.
- updateHotelListing.php - A PHP script for handling the update of a hotel's status (active or inactive).
- delete_hotel.php - A PHP script for handling the deletion of a hotel.

Tools and Technologies

The following tools and technologies have been used to implement the hotel display feature:

- PHP: A server-side scripting language used for web development. PHP has been used to create the scripts responsible for handling the hotel display and management functionality.
- HTML/CSS: The standard markup and styling language for creating web pages. HTML and CSS have been used to structure and style the hotel display page.

- MySQL: A relational database management system for storing and managing the hotel data. MySQL has been used as the backend storage for the hotel data.

Workflow

- An administrator logs into their GoFly account.
- The administrator navigates to the "Hotels" page by clicking the "Hotels" option on the top navigation bar.
- The administrator can view all the hotels listed on the platform, including their name, location, status, and action buttons.
- The administrator can update a hotel's status (active or inactive) by clicking the "Activate" or "Deactivate" button next to the hotel's name. This triggers the hotel_status.php script to update the hotel's status in the MySQL database and displays a confirmation message to the administrator.
- The administrator can delete a hotel by clicking the "Delete" button next to the hotel's name.
- This triggers the delete_hotel.php script to remove the hotel from the MySQL database and displays a confirmation message to the administrator.

Displaying Hotel Details

- The admin_hotelDisplay.php script retrieves the hotel data from the MySQL database and displays them on the "Hotels" page.
- The displayed hotel data includes the hotel name, location, status (active or inactive), and action buttons (activate, deactivate, delete).
- The script also allows administrators to sort the displayed hotels based on the hotel name, location, or status.

Updating Hotel Status

- The updateHotelListing.php script receives the updated hotel status (active or inactive) from the administrator and updates the corresponding hotel's status in the MySQL database.
- The script also displays a confirmation message to the administrator after a successful update.

Deleting a Hotel

- The delete_hotel.php script receives the request to delete a hotel from the administrator and removes the corresponding hotel from the MySQL database.
- The script also displays a confirmation message to the administrator after a successful deletion.

Flight Display Software Documentation

Overview

This document provides an overview of the flight display feature for the GoFly platform. It covers the components and functionality required to allow the admin to view flight listings. The feature aims to help the admin make informed decisions by providing them with all necessary information about the available flights.

Components

- displaylist.css - A CSS file containing the styling for the flight display components.
- admin_displaylist.php - PHP script for displaying the list of flights.

Tools and Technologies

The following tools and technologies have been used to implement the flight display feature:

- PHP: A server-side scripting language used for web development. PHP has been used to create the scripts responsible for displaying the flight listings.
- HTML/CSS: The standard markup and styling language for creating web pages. HTML and CSS have been used to structure and style the flight display.
- MySQL: A relational database management system for storing and managing the flight data. MySQL has been used as the backend storage for the flight data.

Workflow

- The admin logs into their GoFly account.

- The admin navigates to the "Flights" page by clicking the "Flights" option on the top navigation bar.
- The admin can view all the available flights with information such as flight number, departure and arrival cities, departure and arrival dates, and ticket price.
- The admin can sort and filter the flights based on criteria such as departure and arrival cities, dates, and price.

Displaying Flight Listings

- The admin_displaylist.php script retrieves the flight data from the MySQL database and displays it on the "Flights" page. The flight listings are displayed in a table format with columns for flight number, departure and arrival cities, departure and arrival dates, and ticket price.
- The admin can sort the flight listings by clicking on the column headers. Clicking on the same header again will toggle between ascending and descending order.
- The admin can filter the flight listings by entering search terms in the search bar located above the flight listings. The search function searches for matches in all columns of the flight listings table.

Future Enhancements

- Add more security to who get make the flight listings. (i.e. only the specific airline can make their listing, and any other airline)

Login Feature Software Documentation

Overview

This document provides an overview of the login feature for the GoFly platform. It covers the components and functionality required to allow registered users to log in to their accounts. The feature aims to authenticate users and provide them with access to the appropriate content based on their user type.

Components

- config.php - A PHP script containing the database connection parameters and other configuration settings required for the application to function properly.
- login.css - A CSS file containing the styling for the login form and the landing page.
- landing.css - A CSS file containing the styling for the landing page.
- login.php - PHP script responsible for handling the login process. This script retrieves the user's login credentials, validates them against the records in the database, and redirects the user to the appropriate page based on their user type.
- landing.php - The landing page for authenticated users who are not administrators. This page provides access to the features and content available to regular users.
- admin_landing.php - The landing page for authenticated users who are administrators. This page provides access to the features and content available only to administrators.

Tools and Technologies

The following tools and technologies have been used to implement the login feature:

- PHP: A server-side scripting language used for web development. PHP has been used to create the scripts responsible for handling the login process and database operations.
- HTML/CSS: The standard markup and styling language for creating web pages. HTML and CSS have been used to structure and style the login form and the landing pages.
- MySQL: A relational database management system used to store and manage user account information.
- password_verify(): A built-in PHP function used to verify the user's input password against the hashed password stored in the database.
- Session: A built-in PHP feature used to store user data across multiple pages and requests.

Workflow:

1. The user navigates to the login.php page and enters their credentials (username and password) in the form.
2. Upon submitting the form, the PHP script checks if both fields have been filled out. If not, an error message is stored in the session variable and the user is redirected back to the login page.
3. The script then queries the MySQL database to check if there is a user with the entered username.
4. If a user is found, the script retrieves the stored hashed password and compares it with the entered password using the password_verify() function.
5. If the password is verified, the user is logged in by storing the username and user type in session variables and redirecting them to either the admin_landing.php or landing.php page depending on their user type.
6. If the password is not verified, an error message is stored in the session variable and the user is redirected back to the login page.

Implementation Details:

1. The script starts by including the config.php file which contains the necessary configuration information for establishing a connection to the MySQL database.
2. The script starts a PHP session using the session_start() function.
3. The script checks if the form has been submitted using the $_SERVER["REQUEST_METHOD"] superglobal variable.
4. If the form has been submitted, the script retrieves the entered username and password using the mysqli_real_escape_string() function to prevent SQL injection attacks.
5. The script checks if both fields have been filled out, and if not, an error message is stored in the session variable and the user is redirected back to the login page using the header() function.
6. The script then prepares a SELECT query to check if there is a user with the entered username using the prepare() function and binds the username parameter using the bind_param() function.
7. The query is executed using the execute() function and the number of rows returned is checked using the num_rows property of the statement object.
8. If a user is found, the script retrieves the stored hashed password and compares it with the entered password using the password_verify() function.
9. If the password is verified, the user is logged in by storing the username and user type in session variables using the $_SESSION superglobal variable and redirecting them to either the admin_landing.php or landing.php page depending on their user type using the header() function.
10. If the password is not verified, an error message is stored in the session variable and the user is redirected back to the login page using the header() function.

11. The script ends by closing the MySQL database connection using the mysqli_close() function.

Future Enhancements

- Add more security such as having a two-factor authentication function.

Signup Software Documentation

Overview

This document provides an overview of the signup feature for the XYZ platform. It covers the components and functionality required to allow users to create a new account and login to their existing account. The feature aims to enable users to access the platform's services and features.

Components

- login.css - A CSS file containing the styling for the signup account components.
- landing.css - A CSS file containing the styling for the signup account components
- signup.php - PHP script responsible for handling the account creation process.
- login.php - PHP script responsible for handling the login process.

Tools and Technologies

The following tools and technologies have been used to implement the signup feature:

- PHP: A server-side scripting language used for web development. PHP has been used to create the scripts responsible for handling the creation of user accounts and login authentication.
- HTML/CSS: The standard markup and styling language for creating web pages. HTML and CSS have been used to structure and style the signup and login forms.

- MySQL: A relational database management system for storing and managing user account information. MySQL has been used as the backend storage for the user data.

Workflow

Creating a New Account

- A user navigates to the "Signup" page by clicking the "Signup" option on the top navigation bar.
- The user fills out the signup form with their personal information, including their full name, email address, and password.
- The "Create Account" button triggers the signup.php script, which processes the submitted information, validates it, and saves it to the MySQL database.
- If the submitted information is valid, the user is redirected to the "Login" page, where they can log in to their newly created account.
- If the submitted information is invalid, the user is presented with an error message indicating the invalid fields that need to be corrected.

Logging In to an Existing Account

- A user navigates to the "Login" page by clicking the "Login" option on the top navigation bar.
- The user fills out the login form with their email address and password.
- The "Log In" button triggers the login.php script, which validates the submitted information and grants the user access to their account.
- If the submitted information is valid, the user is redirected to their account dashboard, where they can access the platform's services and features.
- If the submitted information is invalid, the user is presented with an error message indicating the incorrect login credentials.

Creating a New Account

- The signup.php script generates the signup form with the required fields: full name, email address, and password.
- The submitted information is validated to ensure that the email address is unique and that the password meets the platform's security requirements.
- If the submitted information is valid, the script saves the new user account data to the MySQL database, and the user is redirected to the login page.
- If the submitted information is invalid, the user is presented with an error message indicating the invalid fields that need to be corrected.

Logging In to an Existing Account

- The login.php script validates the submitted email address and password against the user account data stored in the MySQL database.
- If the submitted information is valid, the script grants the user access to their account and redirects them to their account dashboard.
- If the submitted information is invalid, the user is presented with an error message indicating the incorrect login credentials.

Future Enhancements

- Add more security such as having a two-factor authentication function.