Complete Website Documentation - Use Case Scenarios

Version 1.0Prepared by: Muhammad Bilal SaleemOrganization: Code HubDate: 23/10/2024

**Table of Contents**

**Introduction**

1.1 Purpose

1.2 Scope

1.3 Intended Audience

1.4 Product Features

**System Features**

2.1 User Authentication

2.2 Course Management

2.3 Enrollment System

2.4 Contact Form Handling

2.5 Website Interactivity

2.6 Admin Content Management

**Functional and Non-Functional Requirements**

3.1 Functional Requirements

3.2 Non-Functional Requirements

**Use Case Scenarios**

4.1 User Visits the Homepage

4.2 User Views Course Details

4.3 User Contacts Support

4.4 User Submits a Form and Receives Confirmation

4.5 Website Loads Styles and Enhancements

4.6 User Interacts with Website Features

4.7 Administrator Updates Website Content

**1. Introduction**

**1.1 Purpose**

This document provides a comprehensive guide to the use case scenarios of the Code Hub website, detailing user interactions, system processes, and possible exceptions. It ensures ease of maintenance and future improvements.

**1.2 Scope**

The documentation includes details on all major website interactions, covering user navigation, form submissions, and administrative tasks, ensuring optimal functionality and usability.

**1.3 Intended Audience**

This document is intended for developers, designers, administrators, and project managers responsible for maintaining and improving the Code Hub website.

**1.4 Product Features**

Homepage Navigation: Users can browse featured courses, testimonials, and other platform details.

**Course Details:** Users can explore course descriptions, pricing, and enrollment options.

**Contact Form:** Users can submit inquiries via a structured form.

**Form Submission Confirmation:** Users receive acknowledgment after submitting forms.

**Website Styling & Enhancements:** The website loads necessary stylesheets and JavaScript enhancements.

**Interactive Features:** Users can interact with dynamic elements like carousels and popups.

**Administrator Management:** Admins can update and deploy content changes.

**2. Use Case Scenarios**

**Use Case 1: User Visits the Homepage**

**Primary Actor:** Visitor

**Actions:**

* The visitor opens the website and lands on index.html.
* The visitor browses featured courses, testimonials, and navigation links.

**Pre-conditions:** The website must be accessible.

**Post Conditions:** The visitor gathers information about the platform.

**Exceptions:** Slow loading due to network issues.

**Use Case 2: User Views Course Details**

**Primary Actor:** Visitor

**Actions:**

* The visitor navigates to courses.html.
* They browse available courses and review pricing and descriptions.

**Pre-conditions:** The course catalog must be updated and accessible.

**Post Conditions:** The visitor identifies a course of interest.

**Exceptions:** Course data fails to load due to a database issue.

**Use Case 3: User Contacts Support**

**Primary Actor:** Visitor

**Actions:**

* The visitor navigates to contacts.html.
* They fill out the contact form and submit it.

**Pre-conditions:** The form must be active and connected to a backend script.

**Post Conditions:** A support request is sent successfully.

**Exceptions:** Submission fails due to missing fields or network errors.

**Use Case 4: User Submits a Form and Receives Confirmation**

**Primary Actor:** Visitor

**Actions:**

* The visitor submits a form (e.g., course inquiry or contact request).
* They are redirected to mail.html, confirming their submission.

**Pre-conditions:** Form submission must be properly configured.

**Post Conditions:** The visitor receives confirmation.

**Exceptions:** Email confirmation fails due to incorrect setup.

**Use Case 5: Website Loads Styles and Enhancements**

**Primary Actor:** Website System

**Actions:**

* The website loads necessary CSS stylesheets (main.css, bootstrap.css, etc.).
* JavaScript enhancements (jquery.nice-select.min.js, waypoints.min.js, etc.) are applied.

**Pre-conditions:** Stylesheets and scripts must be properly linked.

**Post Conditions:** The website is fully styled and interactive.

**Exceptions:** Broken links or missing files cause UI inconsistencies.

**Use Case 6: User Interacts with Website Features**

**Primary Actor:** Visitor

**Actions:**

* User interacts with carousels (owl.carousel.min.js).
* They use search functionality and popups (jquery.magnific-popup.min.js).

**Pre-conditions:** The scripts must be loaded correctly.

**Post Conditions:** The visitor successfully interacts with website elements.

**Exceptions:** JavaScript errors prevent interaction.

**Use Case 7: Administrator Updates Website Content**

**Primary Actor:** Administrator

**Actions:**

* The administrator edits HTML, CSS, or JavaScript files.
* Changes are deployed to the server.

**Pre-conditions:** The administrator has access to website files.

**Post Conditions**: The updated content is live.

**Exceptions:** Deployment fails due to file permission errors.

**System Features:**

**1.User Authenticatio**n

* Secure login and registration system for users and administrators.
* Password encryption for security.

**2.Course Management**

* Add, update, and delete courses.
* Categorization and filtering of courses.

**3.Enrollment System**

* Users can enroll in courses.
* Enrollment records are stored for future access.

**4.Contact Form Handling**

* Users can submit inquiries through a form.
* Messages are stored and forwarded to support.

**5. Website Interactivity**

* Carousels, search functionality, and popups for an enhanced user experience.
* Interactive elements powered by JavaScript libraries.

**6.Admin Content Management**

* Administrators can edit, update, and publish content.
* Secure access control for admin panel functionalities.

**Functional and Non-Functional Requirements**

**1. Functional Requirements**

* The system shall allow users to register and log in securely.
* Users shall be able to browse and enroll in available courses.
* The system shall provide a contact form for user inquiries.
* Users shall receive confirmation upon successful form submission.
* The administrator shall manage website content, including adding and updating courses.
* The website shall support interactive elements like search functionality, carousels, and popups.
* User activity logs shall be maintained for security and analytics purposes.

**2. Non-Functional Requirements**

**Performance:** The website should load within 3 seconds under normal conditions.

**Scalability:** The system should handle increasing numbers of users and courses.

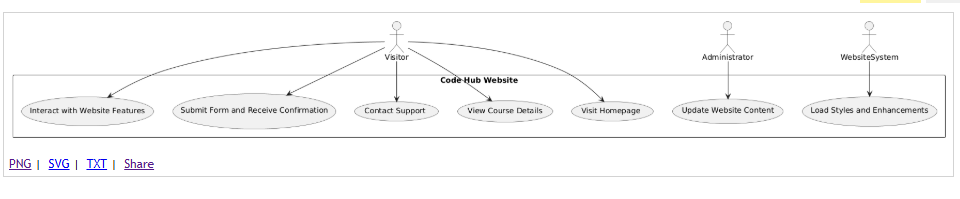
**Security:** User authentication should be implemented using encrypted passwords.

**Usability:** The interface should be intuitive and user-friendly.

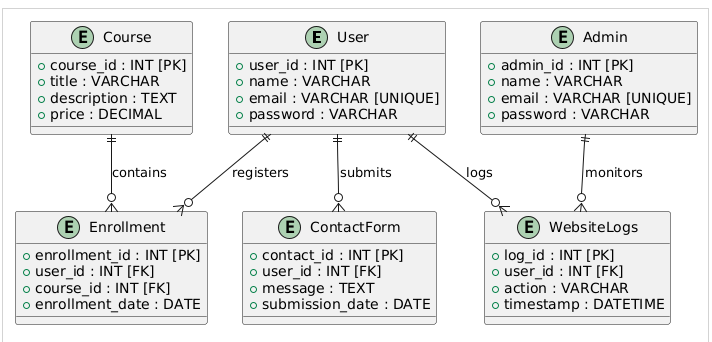
**Availability:** The system should maintain 99.9% uptime with failover mechanisms.

**Compatibility:** The website should be accessible across modern browsers and devices.

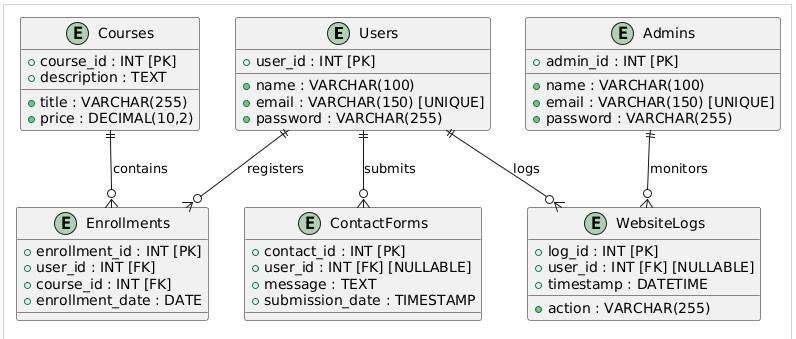
**Use Case Daigram:**



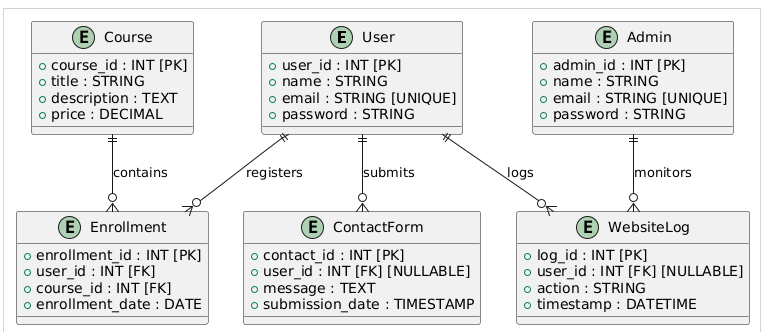
**ERD Diagram:**

****

**Physical Model Diagram:**

****

**Logical Model Diagram:**

****