1. Problem Research

1.1. General Introduction

Website Origin:

The Alpine Ascents Vietnam website was established with the goal of providing unique adventure travel experiences at renowned destinations in Vietnam. Building on the expertise of Alpine Ascents International and designed to harmonize with the natural beauty and rich cultural heritage of Vietnam, Alpine Ascents Vietnam offers premier exploration experiences, from conquering the majestic Fansipan peak and exploring the magnificent cave systems of Phong Nha - Kẻ Bàng, to enjoying the mystical beauty of Bà Nà, and embarking on exciting trekking adventures at Núi Chúa.

Need for Developing This Website:

In the context of the growing adventure tourism industry in Vietnam, there is an increasing demand for a professional and comprehensive online platform for searching, booking, and participating in exploration trips. The Alpine Ascents Vietnam website serves not only as a channel to showcase adventure tours but also as a resource for information, knowledge, and community for adventure travel enthusiasts. Establishing this website will facilitate easier customer access, enhance brand reputation, and create a competitive edge in the market.

1.2. Current Website System

Currently, there are several companies in Vietnam providing adventure travel and exploration services, such as Oxalis Adventure, Viettrekking, and Adventure Vietnam.

Strengths of Competitors:

Intuitive and user-friendly interface.

Online tour introduction and booking functionality.

Detailed information about trips and necessary equipment.

Weaknesses of Competitors:

Lack of rich, in-depth blog content about real experiences.

Interface is not fully user-friendly across all devices.

Limited customer interaction and no online community.

1.3. Proposed System Functions

1.3.1. Adventure Tour Programs

a. Description:

Display a list of available adventure tours, including tour name, price, images, and a brief description of each tour.

Each tour detail will include comprehensive information such as location, requirements, itinerary, necessary equipment, and important participation notes.

Customers can view tour information and proceed to book a tour if interested.

b. Functions:

Tour list

Tour detail page with detailed information and images.

"Book Tour" button that redirects to the booking form.

1.3.2. Management of Adventure Tours

a. Description:

Admin can manage the list of adventure tours, including adding, editing, or deleting tours.

Admin can update pricing, images, and other detailed information about the tours.

Admin has the authority to cancel tours.

b. Functions:

CRUD (Create, Read, Update, Delete) operations for adventure tours.

Update pricing.

Manage images, descriptions, and detailed information about tours.

1.3.3. Gallery

a. Description:

Customers can view a library of images and videos related to tours, organized by each adventure tour.

The library is divided into individual albums for each tour, allowing users to view photos and videos related to specific tours.

b. Functions:

Display a list of albums corresponding to each tour.

Support slideshow viewing mode.

1.3.4. Gallery Management

a. Description:

Admin has the authority to manage the image and video library for each tour.

Admin can add, update, or delete images and videos related to the adventure tours.

Manage directories or albums by each tour.

b. Functions:

Add, delete, and update photos and videos for each tour.

Create and manage albums (folders) for each tour.

1.3.5. Branches

a. Description:

Display a list of the company's branches in various countries or regions.

Each branch will have information about the address, phone number, and location on a map.

Integrate a Maps API to allow users to view the exact location of each branch.

b. Functions:

Display branch information such as name, address, and phone number.

View the branch location on the map.

1.3.6. Branch Management

a. Description:

Admin has the authority to add, delete, and edit branch information.

Manage detailed information for each branch, including name, address, phone number, and geographical location.

b. Functions:

CRUD (Create, Read, Update, Delete) operations for branch information.

Update location coordinates (latitude, longitude) for the map.

1.3.7. Information

a. Description:

Provide general information about mountaineering: history, classification, techniques, etc.

Information may include an introduction to the company, mission, core values, and contact details.

b. Functions:

Display static information about the company.

Serve as a resource for researching and accessing supportive information about mountaineering.

1.3.8. User Account Management

a. Description:

Users can register, log in, and manage their accounts.

Users can edit personal information such as name, phone number, email, and password.

b. Functions:

Register, log in, and log out of accounts.

Manage personal information and update details.

1.3.9. Admin Account Management

a. Description:

Admin has the authority to manage account information for all users in the system.

Manage permissions between admin accounts and regular users.

Admin can add or delete user accounts, update information, and reset passwords.

b. Functions:

CRUD operations for user accounts (add, delete, edit).

User role management (admin or user).

Manage account status (active or inactive).

1.3.10. Order Placement

a. Description:

Users can place tour bookings through an order form.

The form includes customer personal information (name, phone number, email) and tour details (tour name, number of participants, total order value).

After placing an order, users receive a notification regarding the order result (success or error).

b. Functions:

Order form with fields for personal information and tour details.

Calculate the total order value based on quantity and tour price.

Store the order in the database and display the result.

1.3.11. Order Management

a. Description:

Admin can view a list of all orders placed by customers.

Admin can view details of each order, including customer information, booked tour, number of participants, and total value.

b. Functions:

Display a list of orders with detailed information.

Update and modify order information.

View details of each order, including user information and booked tour.

1.4. Limitations of the System

No Support for User Avatar Updates: Currently, the system does not have a feature that allows users to upload or change their avatars.

No Order History Tracking: The system does not provide functionality for tracking users' order history and bookings.

No Online Payment Integration: The system currently lacks online payment capabilities.

Limited User Search Functionality: The system may not support searching for users or administrators based on specific criteria such as name, email, or account status.

No Shopping Cart Feature: The system does not provide a shopping cart feature to allow users to add multiple tours and process payment for all at once.

No Order Filtering by Criteria: The system lacks the ability to filter orders by status, order date, or other criteria.

No Handling of Spam Orders: The system does not have mechanisms to detect and manage spam or invalid orders.

No Order Status Management: The system does not support updating or changing the status of orders (e.g., pending confirmation, paid, completed).

No Tour Filtering by Criteria: The system does not allow users to filter tours based on criteria such as location, price, duration, or tour type.

No Branch Filtering by Geographical Locations: The system does not provide functionality to filter branches by geographical location.

No Reporting and Analytics Features: The current system does not offer reports or analytics on sales, number of tours sold, or user activity.

No Detailed Role-Based Permissions: The system does not support detailed permissions for admins or specific roles (e.g., an admin managing only the gallery or orders).

Not Tested with Over One Million Records: The system has not been performance tested with a large volume of records.

No Multilingual Support: The system currently only supports a single language, lacking the ability to switch between different languages.

No Advanced Security Features (2FA, Data Encryption): The system does not integrate advanced security methods such as two-factor authentication (2FA) or encryption of personal data and transactions.

No Notification System: The system does not support sending notifications via email or push notifications for important events such as successful orders, order cancellations, or status updates.

No Feedback and Rating System: The system does not provide functionality for users to give feedback or rate the tours they have participated in.

No Promotion and Discount Management System: The current system does not have management features for promotions or discount codes for users when booking tours.

No Error Logging and Analysis System: The system lacks the ability to store and analyze error logs for quick detection and resolution of issues.

1.5. Hardware and Software Requirements

Hardware:

Intel Core i3/i5 Processor or higher

8 GB RAM or above

Color SVGA

500 GB Hard Disk space

Mouse

Keyboard

Software:

Operating System: Windows

Browsers: Edge, Chrome, Mozilla Firefox, Safari

2. Requirements Analysis

2.1. Users of the System

Customers: Individuals interested in adventure trips, trekking, and mountaineering. They use the website to search for information, book tours, view the photo gallery, check the list of branches worldwide, and research essential information related to mountaineering. They can also manage their personal information when registering and logging into the system.

Administrators: Individuals who manage the website, add and update tour information, process bookings, manage the photo gallery, oversee partner lists, and handle customer information.

2.2. System Functions

| **Function** | **Input Information** | **Output Information** | **Processing Method** | **Data to be Stored** |
| --- | --- | --- | --- | --- |
| **Tours** | - List of tour programs, including tour name, description, price, itinerary, and other special information. | - List of available tours with detailed descriptions and images. - Options for online booking. | - Retrieve data from the database about tours. - Display detailed information about each tour. - Provide options for booking tours. | - Information about tour programs (name, description, price, itinerary). - Images and videos related to the tours. |
| **Tour Management** | - Tour information: tour name, price, representative image, location, highlights, best time to visit, directions, trekking routes, required items, warnings, participation requirements for customers. | - List of all tours, displaying details of each tour (name, price, location, etc.). - Details of a specific tour when requested by the user. - Result of adding, editing, or deleting a tour (success or error). | - User (admin) submits a request to add, edit, or delete tour information. - The system checks input data: all required fields (name, price, location, etc.) must be fully filled. - If the data is valid, the system performs operations with the database (add, edit, delete). - The system returns results to the user interface. | - Tours table: stores detailed information about each tour. - Data fields: id, name, price, image, location, features, best time, directions, trekking routes, items, cautions, requirements, created\_at, updated\_at. |
| **Gallery** | - Images, videos, and links (e.g., YouTube) related to each tour. | - Page displaying images and videos for each tour. - Display images in a slideshow format for each tour. | - When a user accesses the gallery page for a tour, the system retrieves the list of images and videos for that tour from the galleries table. | - Images (stores the image paths). - Videos (stores links to YouTube videos). - tours\_id (links to the specific tour). |
| **Gallery Management** | - List of images and videos for each tour. - Tour ID linked to the gallery. - URL or YouTube link for videos. | - List of galleries corresponding to each tour. - List of images and videos belonging to each tour. - Result of adding or removing images/videos in the gallery. | - When a user accesses the gallery page for a tour, the system retrieves the list of images and videos for that tour from the galleries table.  - If the admin needs to add or delete images or videos, they will submit a request through the admin interface.  - The system checks the validity of the data (images must be in a valid format, videos must have valid links).  - Update the galleries table and store the new information. | - Galleries table: stores links to images and videos for each tour. - Data fields: id, tours\_id, images (stored as JSON containing a list of image links), videos (stored as JSON containing a list of video links), created\_at, updated\_at. |
| **Branches** | Branch name, address, phone number, geographical location. | List of branches along with a map displaying the location of each branch. | - Retrieve branch information from the database. - Use an API to display the locations of branches on the map. | name, address, phone, positioning |
| **Branch Management** | - Branch information: name, address, phone number, positioning information. - Admin requests to add, edit, or delete branches. | - List of the company's branches worldwide. - Details of a specific branch. - Result of adding, editing, or deleting a branch (success or error). | - User (admin) submits requests to add, edit, or delete branch information.  - The system checks required fields (name, address, phone number). - The system performs operations with the database (add, edit, or delete). - Returns results to the admin. | - Branches table: stores branch information.  - Data fields: id, name, address, phone, positioning, created\_at, updated\_at. |
| **Information** | Information related to history, types, techniques, accommodation, and dangers of mountaineering and trekking. | Display articles and information categorized accordingly. | - Store related information as articles (blogs). - Classify and display by specific categories. | - Data fields: title, content, category, status. |
| **Account Management** | Account information upon registration: name, phone number, email, password.   * When updating information: avatar, account status (status), verification code (verification\_code). * Login information: email, password. * Information required for password recovery, if any. | * User account information (name, email, avatar, role). * Login status: success or failure. * Result of registration, account update, or password change. * Verification or account recovery notifications, if any. | * When a user registers an account, the system checks if the email and phone number already exist. * If valid, create a new account and store the information in the database. * The system will send a verification code via email; if verified correctly, the account status will change to active. * When a user logs in, the system checks the email, password, and returns the login result. * When a user updates information, the system verifies access rights and saves changes to the database. * If a user requests password recovery, the system will send a verification code via email. | Users table: stores user account information.  Data fields: id, name, phone, email, password, avatar, role, verification\_code, status, created\_at, updated\_at. |
| **User Management** | * User information such as name, email, phone number, avatar, password, role. | * Display the list of users, details of each user. | * Create, update, delete user information. * Manage access rights (admin, regular user). | name, email, phone, password, avatar, role |
| **Order** | * Customer information when placing an order: name, phone number, email, notes (if any). * Information about the booked tour: tour ID, quantity, total order value. | * List of orders placed. * Detailed information about a specific order (customer name, booked tour, quantity, order value). * Result of the order placement (success or error). | * When a user places an order, the system retrieves tour information, quantity, and calculates the total order value. * The user enters personal information. * The system confirms the order and stores it in the database. * The system returns the order placement result. | Orders table: stores information about the order.  Data fields: id, user\_id (user ID), tour\_id (tour ID), name (customer name), phone, email, quantity (number of tours), total (total value), note, created\_at, updated\_at. |