

***Documented Report On Project***

Course Code :SEN2241

Course Title : OBJECT OREINTED DESIGN AND ANALYSIS AND IMPLEMENTATION

Group Number :21

Project Topic: CAMTOUR

Link to GitHub Repository :

Group Leader : Gnowa Temga Gesse Kevin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SN | NAME | MATRICULE | TEAM ROLE | %PARTICIPATION |
| 1 | GNOWA TEMGA GESSE KEVIN | ICTU20233723 | Scrum Master | 25% |
| 2 | ASOBO KHAN DUGA | ICTU20233920 | Coder | 15% |
| 3 | JOHN PAUL NJUH AWA | ICTU20233898 | Coder | 15% |
| 4 | AYUK ETANG JUNIOR FAVOR | ICTU20233751 | Coder | 15% |
| 5 | FAKAM FANKAM BLAISE | ICTU20234199 | Coder | 15% |

***PROJECT: CAMTOUR(CAMEROON TOURISTIC WEBSITE)***



***INTRODUCTION***

CamTour is an interactive web platform designed to showcase and facilitate authentic travel experiences across the diverse landscapes and cultures of Cameroon. Built using modern web technologies such as Next.js, React, and Tailwind CSS, CamTour provides detailed information on tours, destinations, and cultural activities, making it an essential companion for travelers seeking immersive and responsible tourism opportunities in Cameroon.

The platform features a rich interface with destination guides, tour listings, detailed itineraries, customer testimonials, and a seamless booking process. It emphasizes sustainability, community impact, and authenticity, aiming to connect travelers with local traditions and natural wonders safely and meaningfully.

The main aim of the CamTour website is to promote and provide access to authentic, sustainable tourism experiences throughout Cameroon by connecting travelers with curated tours, local guides, and culturally rich destinations. The platform strives to enhance tourism's positive impact on local communities and the environment while offering travelers memorable, well-organized travel opportunities.

Camtour has the following objectives

- Showcase Diverse Destinations: Present detailed information about Cameroon’s coastal regions, highlands, savannas, rainforests, and cultural hubs, highlighting their unique features and attractions.

- Offer Curated Tours: Provide a comprehensive listing of tours and excursions, catering to various interests such as adventure, culture, wildlife, and relaxation.

- Enhance User Experience: Craft an intuitive, modern, and responsive user interface that facilitates easy discovery, filtering, and booking of tours.

- Support Sustainability: Promote environmentally responsible tourism and community involvement through responsible tour offerings.

- Facilitate Booking and Authentication: Implement user signup and login functionalities to enable secure booking and personalized experiences.

- Deliver Rich Content: Supply insightful narratives including historical context, practical information (weather, tips, transport), and customer reviews to educate and inform travelers.

- Ensure Accessibility and Performance: Build an accessible, performant, and responsive web application adhering to modern web design principles.

***Problem Statement***

Tourists interested in exploring Cameroon often face challenges in accessing authentic and reliable information about local destinations, cultural experiences, and available tours. There is a scarcity of platforms that offer well-curated, sustainable, and user-friendly services tailored to highlight Cameroon’s natural beauty and culture responsibly. Additionally, travelers may encounter difficulties navigating fragmented resources, confusion over logistic details, and lack of trust in tour providers.

CamTour addresses these issues by centralizing detailed, trustworthy information about Cameroon's travel opportunities within a cohesive web platform. It facilitates easier exploration and booking of tours while assuring sustainable practices and community benefits. This provides significant value to travelers, tour operators, and local communities alike by fostering sustainable tourism growth in Cameroon.

***Software Development Methodologies***

Software development methodologies are structured approaches to planning, executing, and managing software development projects. They provide frameworks that guide teams in organizing their work, managing resources, and ensuring quality. Common methodologies include:

1. Waterfall Model: A linear and sequential approach where each phase must be completed before the next begins. It is straightforward but inflexible to changes.

2. Agile Methodology: An iterative approach that emphasizes flexibility, customer collaboration, and rapid delivery of functional software. Agile methodologies include Scrum, Kanban, and Extreme Programming (XP).

3. Scrum: A subset of Agile, Scrum focuses on delivering small, incremental changes through time-boxed iterations called sprints. It involves roles like Scrum Master and Product Owner and ceremonies like daily stand-ups and sprint reviews.

4. Kanban: Another Agile methodology that visualizes work in progress and limits the amount of work in each stage of the process. It emphasizes continuous delivery without overloading team members.

5. Extreme Programming (XP): A methodology that emphasizes technical excellence and frequent releases in short development cycles, promoting customer satisfaction and adaptability.

***Comparison Between Different Software Development Methodologies***

|  |  |  |  |
| --- | --- | --- | --- |
| ***Methodology*** | ***Characteristics*** | ***Advantages*** | ***Disadvantages*** |
| Waterfall | Linear,sequential phrases | Simple to understand, easy to manage | Inflexible to changes, late testing |
| Agile | Iterative,Flexible | Adaptable to changes, customer-focused | Inflexible to changes, late testing |
| Scrum | Time-boxed sprints, Defined Roles | Clear roles, regular feedback, quick delivery | Requires commitment, can be mismanaged |
| Kanban | Visual workflow, contionous delivery | Flexible, easy to implement | Requires commitment, can be mismanaged |
| XP | Emphasizes techinal practices | High quality, customer satisfaction | Requires skilled developers, can be resource-intensive |

***Reason for the Choice of Scrum Methodology***

The choice of Scrum methodology for the CamTour project is driven by several factors:

1. Iterative Development: Scrum allows for iterative development, enabling the team to deliver functional components of the website in short cycles (sprints). This is crucial for a project that may require frequent adjustments based on user feedback.

2. Flexibility and Adaptability: The tourism industry is dynamic, and requirements may change based on market trends or user needs. Scrum’s flexibility allows the team to adapt to these changes quickly.

3. Enhanced Collaboration: Scrum promotes collaboration among team members through daily stand-ups and sprint reviews, ensuring that everyone is aligned and can address issues promptly.

4. Customer Involvement: Regular feedback from stakeholders and potential users can be integrated into the development process, leading to a product that better meets user expectations.

5. Focus on Deliverables: The emphasis on delivering small, incremental updates helps maintain momentum and keeps the team motivated by showcasing progress regularly.

In the context of the CamTour project, several related concepts are essential to understand:

1. User -Centered Design: The project focuses on creating a user-friendly interface that caters to travelers' needs, ensuring that the design is intuitive and accessible.

2. Sustainable Tourism: The website promotes sustainable tourism practices, which is a growing trend in the travel industry. This aligns with the project’s mission to support local communities and preserve the environment.

3. Responsive Web Design: Given the diverse user base, the website is designed to be responsive, ensuring a seamless experience across devices, from desktops to mobile phones.

1. Continuous Integration/Continuous Deployment (CI/CD): Implementing CI/CD practices can enhance the development process by automating testing and deployment, ensuring that new features are delivered quickly and reliably.

***Related Literature***

1. Agile Development in Tourism: Literature suggests that Agile methodologies, particularly Scrum, are increasingly adopted in the tourism sector to enhance responsiveness to market changes and customer feedback (Source: Agile in Tourism: A Review of Literature).

2. Sustainable Tourism Practices: Research highlights the importance of integrating sustainability into tourism platforms, emphasizing the role of technology in promoting eco-friendly practices (Source: Sustainable Tourism: A Technological Perspective).

3. User Experience in Travel Websites: Studies indicate that user experience significantly impacts customer satisfaction and conversion rates in travel websites. A focus on intuitive navigation and clear information presentation is essential (Source: User Experience in Travel Websites: A Study).

4. Impact of Technology on Tourism: The literature discusses how technology, including web platforms, has transformed the tourism industry by providing travelers with more information and options, leading to informed decision-making (Source: The Role of Technology in Modern Tourism).

This structured overview provides a comprehensive understanding of the software development methodologies relevant to the CamTour project, the rationale for choosing Scrum, and the context of related concepts

***Research Methodology***

The research methodology for the CamTour project involves a combination of qualitative and quantitative approaches. The qualitative aspect includes user interviews and surveys to gather insights on traveler preferences and pain points in the tourism experience. The quantitative aspect involves analyzing data from existing tourism platforms to identify trends and user behavior. This mixed-methods approach ensures a comprehensive understanding of user needs and market demands, guiding the development of features and functionalities in the CamTour platform.

***System Requirements***

**Functional Requirements**

1. User Authentication: Users should be able to sign up, log in, and manage their accounts.

2. Tour Listings: The system must display a list of available tours with details such as title, price, duration, and location.

3. Search and Filter: Users should be able to search for tours and filter results based on categories, price, duration, and ratings.

4. Booking System: Users must be able to book tours, select dates, and specify the number of travelers.

5. User Reviews: Users should be able to leave reviews and ratings for tours they have experienced.

6. Responsive Design: The website must be accessible and functional on various devices, including desktops, tablets, and smartphones.

**Non-Functional Requirements**

1. Performance: The website should load within 3 seconds to ensure a smooth user experience.

2. Scalability: The system must handle an increasing number of users and tours without performance degradation.

3. Security: User data must be protected through encryption and secure authentication methods.

4. Usability: The interface should be intuitive and user-friendly, allowing users to navigate easily.

5. Availability: The system should have a high uptime percentage, ensuring users can access it at any time

***System Design***

**System Architecture(HLD)**

The architecture of the CamTour system follows a multi-tier architecture:

1. Presentation Layer: This layer consists of the user interface built with React and Tailwind CSS, providing a responsive and interactive experience.

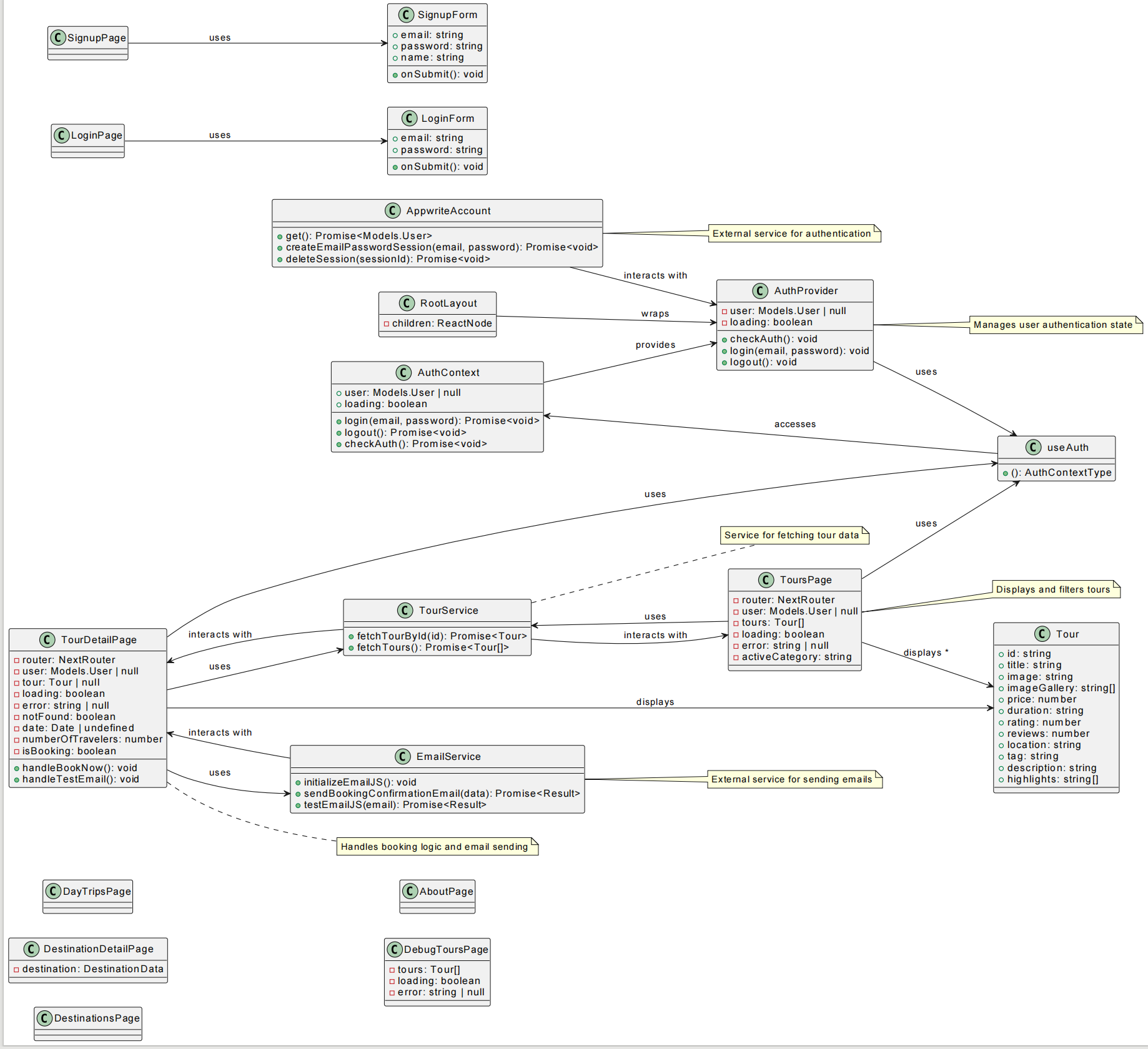
2. Application Layer: This layer handles business logic, user authentication, and data processing. It is built using Next.js, which facilitates server-side rendering and API routes.

3. Data Layer: This layer includes the database (e.g.Appwrite) where user data, tour information, and reviews are stored. It interacts with the application layer through APIs.

***UML Diagrams***

- Use Case Diagram: Illustrates the interactions between users and the system, highlighting key functionalities such as booking tours, leaving reviews, and managing accounts.

- Class Diagram: Represents the structure of the system, showing classes such as User, Tour, Review, and their relationships.



- Sequence Diagram: Demonstrates the flow of operations during a booking process, detailing interactions between the user interface, application logic, and database.

***Application of Scrum***

**Team Organization**

- The Scrum team consists of cross-functional members, including:

- Product Owner: Responsible for defining the product vision and managing the product backlog.

- Scrum Master: Facilitates the Scrum process, removes impediments, and ensures adherence to Scrum practices.

- Development Team: Composed of developers, designers, and testers who collaboratively work on delivering increments of the product.

**Workflow Management**

- The team follows a structured workflow with defined roles and ceremonies:

- Sprint Planning: The team collaborates to plan the work for the upcoming sprint, selecting items from the product backlog.

- Daily Stand-ups: Short daily meetings to discuss progress, challenges, and plans for the day.

- Sprint Review: At the end of each sprint, the team demonstrates completed work to stakeholders for feedback.

- Sprint Retrospective: A meeting to reflect on the sprint, discussing what went well, what could be improved, and actionable steps for future sprints.

**Conflict Resolution**

- Conflicts are addressed through open communication and collaboration. The Scrum Master facilitates discussions to ensure all team members have a voice and work towards a consensus. If necessary, external mediation may be sought to resolve significant disputes.

Challenges Encountered and How we Overcame Them

- Challenge: Initial resistance to adopting Scrum practices among team members.

- Solution: Conducted training sessions to educate us on Scrum principles and benefits, fostering a culture of collaboration and adaptability.

- Challenge : Difficulty in estimating the effort required for user stories.

- Solution: Implemented planning poker sessions to improve estimation accuracy through collective input and discussion.

**Scrum Artifacts**

***Product Backlog***

- The product backlog is a prioritized list of features, enhancements, and bug fixes that need to be addressed. It is continuously refined based on stakeholder feedback and changing requirements.

Product Backlog – Authentication Module

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | User Story | Priority | Effort (Story Points) | Acceptance Criteria | Notes |
| PB-1 | As a user, I want to sign up with my email and password | High | 5 | - Signup form visible - Validates inputs - Shows success/error messages - Creates new user | UI + form validation |
| PB-2 | As a user, I want to log in using email and password | High | 5 | - Login form present - Authenticates user - Redirects on success - Shows error on failure | Integrates with AuthContext |
| PB-3 | As a user, I want to see loading feedback during authentication | Medium | 2 | - Spinner/indicator shows while loading - No flickering or early transitions | Use loading from context |
| PB-4 | As a user, I want to log out safely and be redirected to login | High | 3 | - Clear session/token - Redirected to login screen - AuthContext.user becomes null | Trigger via logout button |
| PB-5 | As a user, I want to be prevented from accessing private pages when logged out | High | 5 | - Redirect to login if not authenticated - Auth guard/higher-order component in place | Add route protection logic |

***Sprint Backlog***

- The sprint backlog consists of items selected from the product backlog for implementation during the current sprint. It includes tasks broken down into actionable items, allowing the team to track progress and manage workload effectively.

Duration: 5 Days

Goal: Implement core functionalities for user authentication, tour management, and email notifications.

Sprint Backlog

|  |  |  |
| --- | --- | --- |
| Sprint | Tasks | Deliverables |
| Sprint 1 | User Authentication: - Create the login form component - Integrate with `AuthContext` for login - Handle login errors - Create the signup form component - Integrate with `AuthContext` for signup - Validate input and handle signup errors | • Functional login and signup pages • Authentication flow integrated with Appwrite |
| Sprint 2 | Tour Management: - Implement `fetchTours` in `TourService` - Create component to list tours - Implement Tour Detail page - Fetch tour details using selected tour ID | • Tours page with available tours • Tour detail page with full information |
| Sprint 3 | Booking Functionality: - Create booking form on Tour Detail page - Validate inputs (date, travelers) - Integrate booking with `EmailService` - Add toast notifications - Redirect unauthenticated users to login | • Booking form on Tour Detail page • Confirmation emails sent on booking |
| Sprint 4 | UI/UX Improvements: - Improve layout and styling - Ensure responsiveness - Implement loading spinners - Add user feedback during async operations | • Polished UI/UX • Loading states added for better user interaction |
| Sprint 5 | Testing and Bug Fixes: - Write unit tests for authentication and tour logic - Review and fix bugs - Ensure all features function as intended | • Solid test coverage • Final stable build ready for deployment |

**1. *Main Purpose***

- CamTour serves as an online portal for users to explore, discover, and book authentic tours and travel experiences in Cameroon. It focuses on providing detailed information about diverse destinations, tours, activities, and cultural experiences throughout the country.

- The site promotes sustainable and responsible tourism while supporting local communities and ecosystems in Cameroon.

***2. Key Features and Functionality***

**a. Home Page**

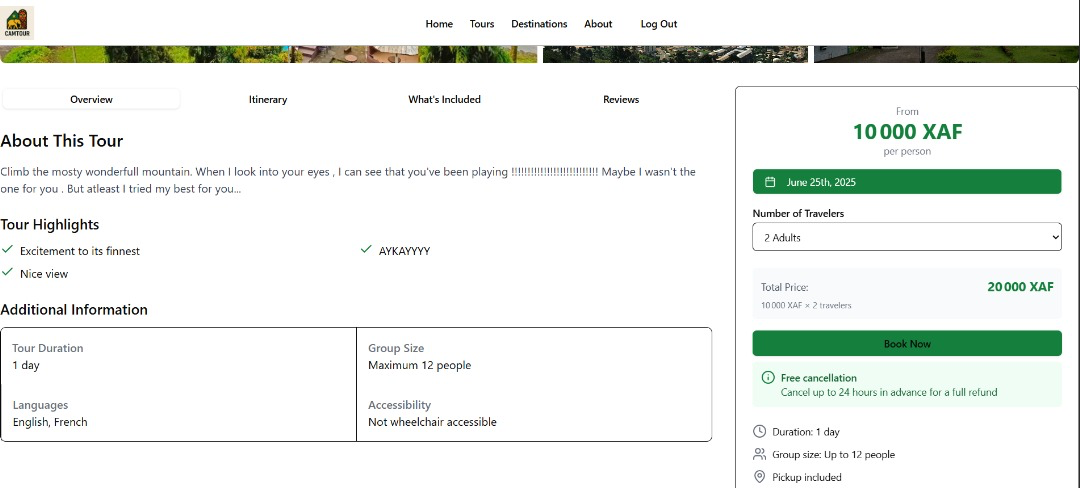
- A visually appealing hero section showcasing the beauty and culture of Cameroon.

- Highlights featured tours to draw user attention to popular or highly rated tours.

- Showcases popular destinations with images and brief info to entice visitors.

- Explains why users should choose CamTour, emphasizing local expertise, authenticity, price guarantee, and small group sizes.

- Newsletter signup to receive travel tips and special offers.



b. **Tour Listings and Filtering**

- Pages dedicated to listing various tours available. Tours cover different categories such as adventure, cultural, wildlife, beach getaways, and hiking.

- Users can browse, filter, and sort tours based on categories, price, rating, duration, and more.

- Each tour listing shows key information including title, location, price, duration, rating, and a thumbnail image.

**c. Tour Details**

- Detailed pages for individual tours that provide:

- Comprehensive descriptions and rich content about what the tour offers.

- High-quality images and an image gallery.

- Tour itinerary with scheduled activities.

- Information on what's included and excluded in the tour price.

- User reviews with ratings and comments.

- Booking options including selecting date and number of travelers.

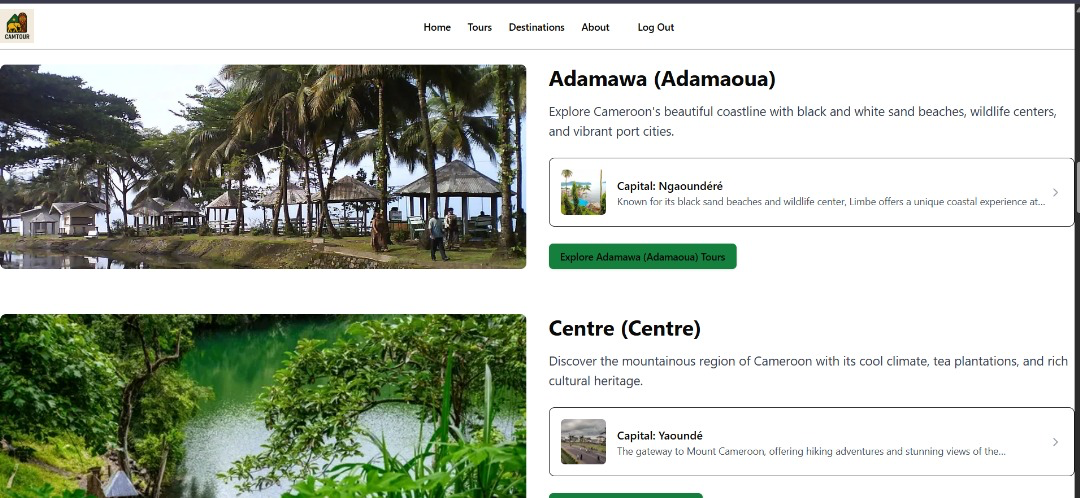
**d. Destination Pages**

- Provides detailed information about different destinations and regions in Cameroon.

- Displays the history, highlights, climate, practical travel tips, and things to do in each region.

- Shows available tours specific to each destination.

- Includes maps and related destinations to help users explore more options.

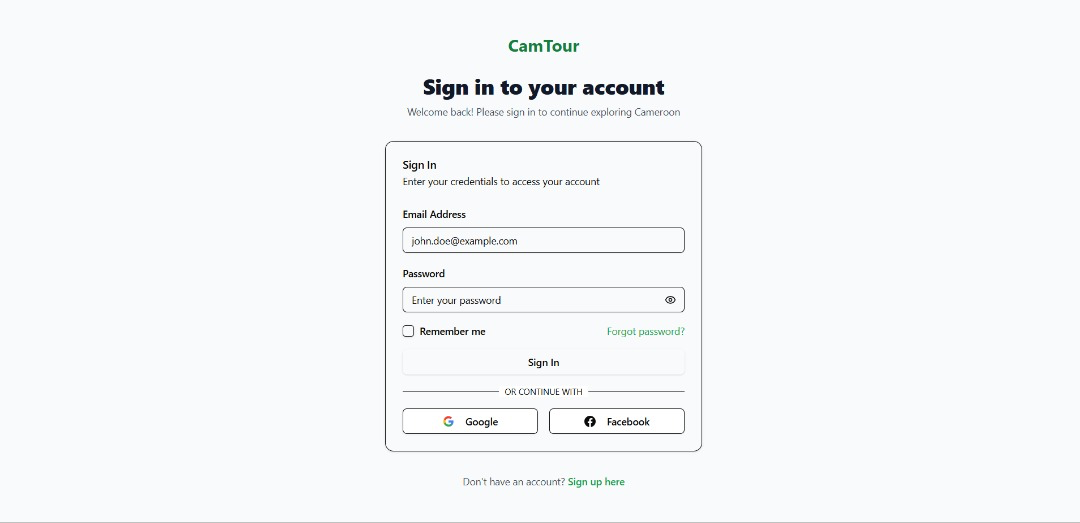


**e. User Authentication**

- Users can create accounts (sign up) or log in to access personalized features.

- Authentication is managed via Appwrite backend services.

- Some routes (like tours) require login, with middleware guarding access.



**f. Additional Utilities**

- Filtering options for day trips and excursions by price, duration, tour type, departure points, and attractions.

- Search bar to quickly find destinations or tours.

- Responsive navigation and UI components optimized for both desktop and mobile.

- Integration with various popular libraries like Lucide React for icons and Radix UI for accessible UI primitives.

***3. Target Audience***

- Individuals interested in travel and tourism in Cameroon.

- Potential tourists looking for authentic, culturally rich travel experiences.

- Local or international travelers seeking detailed, reliable information about Cameroonian destinations.

- Users who want an easy-to-use platform to browse, get detailed info on, and book tours.

4***. Content Focus***

- Authentic Experiences: Emphasizes genuine cultural exchanges and sustainable tourism.

- Wide Variety of Tours: From wildlife safaris and hiking to beach getaways and cultural tours.

- Local Community Support: Promotes tours that benefit local businesses, guides, and communities.

- Detailed Information: Rich content about each destination and tour, including logistics, tips, and customer reviews.

***5. User Flow***

1. User lands on the homepage with visual highlights and quick access to popular tours and destinations.

2. They can navigate to destinations or tours pages to explore specific regions or tours.

3. On tours page, they can filter by category or other criteria and see detailed information.

4. User can sign up or log in to proceed to book tours or access exclusive content.

5. Interactive components like search bar, filters, and tabs help users find relevant information easily.

6. Detailed pages allow users to understand the itinerary, pricing, what's included, and read reviews.

7. Users can subscribe to the newsletter for updates.

***6.Technology Stack***

- Built with Next.js (React framework) for fast server-side and static rendering.

- Styled with Tailwind CSS for modern and well-organized design.

- Uses Appwrite as the backend service for authentication and data storage (such as tours).

- Utilizes various UI components from Radix UI for accessibility and composability.

- Uses TypeScript and React hooks for state and data management.

- Uses client-side routing and dynamic pages to display specific tour and destination details.

***Programming Languages Used and Code Functions***

1. ***Programming Languages Used***

- TypeScript/JavaScript: The bulk of the project consists of React components written in TSX/JSX format for UI rendering and client-side logic. These include pages, components, utilities, context providers, and hooks.

- CSS (Tailwind CSS): Styling is done using Tailwind CSS classes integrated in components plus a globals.css file for additional custom CSS variables and global styles.

- JSON: Used for configuration files such as launch.json for debugging configurations.

- XML: Used for IDE/project metadata and configuration files such as .iml, .xml files in the .idea folder.

- Binary files for fonts and images (e.g., .woff font files, .ico, .png files) which are assets used in the UI.

2***. Key Code Files and Their Functions***

- pages/ (multiple page.tsx files)

- Various Next.js page components implementing key routes:

- Home page with hero section, featured tours, popular destinations, newsletter signup.

- About page describing the company story, values, team, testimonials, and partners.

- Tours pages with filters, listing, details and debugging support.

- Destination detail pages with regions, highlights, activities, tours, maps, and practical info.

- Login and Signup pages with forms for authentication.

- Day trips page with filters and tour listings.

- Each page uses React, Next.js routing conventions and imports UI components.

- components/

- UI components for cards, buttons, inputs, modals, dropdowns, tabs, accordions, avatars, badges, checkboxes, dialogs, drawers, paginations, popovers, switches, tables, toasts, toggles, etc.

- Reusable design system components styled with Tailwind CSS.

- Components for layout structure including header navigation and footer.

- contexts/

- AuthContext provider for managing user authentication globally.

- lib/

- Service functions such as fetchTours and fetchTourById to interact with backend APIs.

- styles/

- globals.css defines CSS variables for theming and base styles using Tailwind CSS.

- hooks/

- Custom React hooks like useToast for toast notification management.

- utils/

- Utility functions for classnames (cn) or similar helper methods.

- Fonts and Images

- Binary font files in .woff format for custom fonts.

- Images used for destinations, tours, logos, and placeholders.

- Config and Metadata Files

- launch.json for IDE debug configurations.

- Project XML files for indexing, workspace, and version control settings.

3***. Overall Architecture and Technologies***

- The project is built with Next.js framework for React-based server-side rendering and routing.

- Styling relies heavily on Tailwind CSS for utility-first CSS.

- Component-driven structure with a collection of modular, reusable UI components.

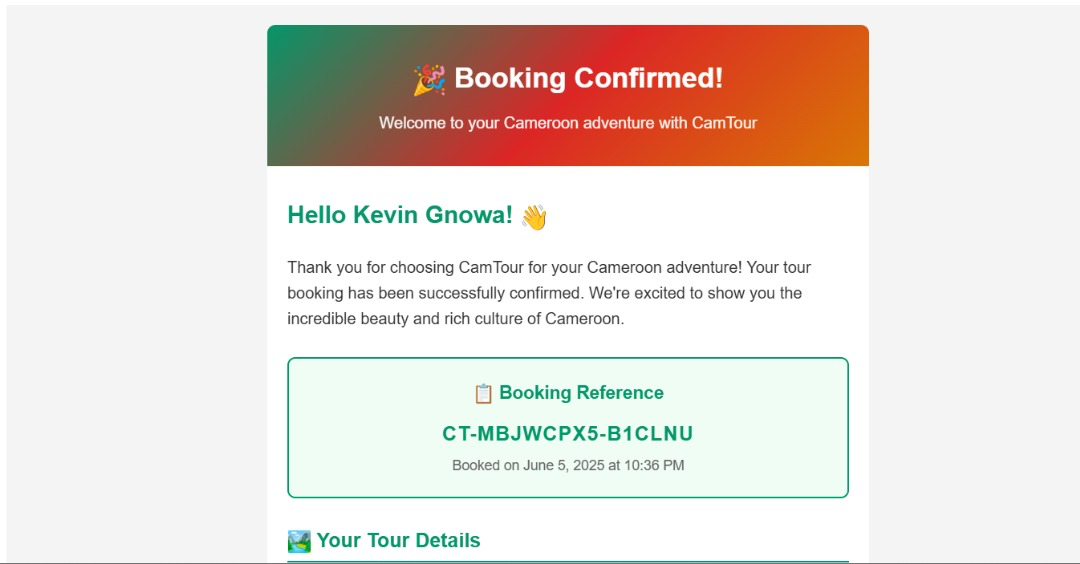
- Supports authentication using Appwrite services as inferred from login and signup forms.

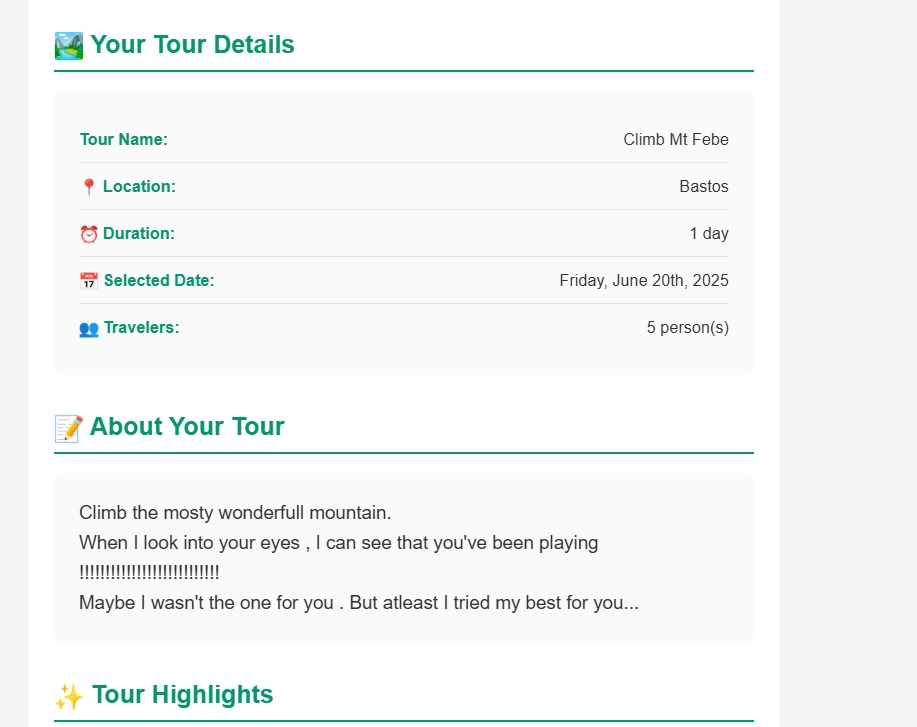
- Uses icon libraries such as lucide-react for consistent iconography.

- Responsive, modern design focusing on a tourism site for Cameroon with rich content about destinations and tours.

***Test Case Document***

Our test subject Gnowa Kevin booked a ticket to go to Mt.Febe Bastos for 1 day on Friday June 20th 2025 with 5 persons. He is supposed to receive a confirmation email about the booking. [screenshot below]





**Proposed Algorithms**

1. Search Algorithm: Implement a search algorithm that filters tours based on user input, utilizing efficient data structures for quick lookups.

2. Booking Algorithm: An algorithm to handle the booking process, ensuring that selected tours are available and managing user data securely.

3. Review Aggregation Algorithm: An algorithm to calculate average ratings and display user reviews effectively.

This structured overview provides a comprehensive understanding of the research methodology, system requirements, design, application of Scrum, and other relevant aspects of the CamTour project.

This is a modern Next.js web application coded primarily in TypeScript/JavaScript using React and Tailwind CSS. It includes multiple pages, reusable UI components, context providers for state management, and service functions for data fetching. Configuration files assist in project setup and tooling. Assets include fonts and image files forming the visual identity of the site. The code is well-structured for a tourism platform showcasing Cameroon travel experiences this website is a comprehensive, user-friendly platform designed to help travelers discover, learn about, and book authentic and diverse travel experiences across Cameroon while maintaining a strong focus on sustainability and community empowerment.