Software Requirements Specification Document

CULTURE HUB, TEAM 1, CS21B044, CS21B043, CS21B003, CS21B031, CS21B006

Note: This is a "living document", meaning its content will change with the implementation of the project. Use it to capture key project requirements and make sure that your product features match the requirements exactly – if you wish to add any features, they must be added first to the requirements. Changes in the document must be approved by the customer (mentor) and the instructor/TA in-charge.

Remove this text and the descriptive paragraphs in each section stating what to do before you turn it in.

Brief problem statement

Our project aims to develop a cultural website to address the issue of limited knowledge and understanding among people about diverse cultures around the world. Many individuals lack exposure to, and awareness of, cultures different from their own. This website seeks to bridge this knowledge gap by providing easily accessible and engaging content that educates and fosters cultural appreciation and understanding among users.

System requirements

Front-End:

- React: The front-end will be developed using React for creating dynamic and responsive user interfaces.
- HTML/CSS: Standard web technologies for structuring and styling the user interface.
- JavaScript: To enhance user interactions and handle client-side functionality.

Back-End:

- Node.js: The back-end will be built using Node.js for server-side scripting.
- Express.js: To create a RESTful API and handle HTTP requests.
- MongoDB: As the database to store content, user data, and other relevant information.

ChatGPT API Integration:

- Integration with ChatGPT APIs to provide conversational features and answer user queries related to different cultures.

Google API Integration:

- Integration with Google Maps API for location-based information and services.
- Integration with Google Translate API to provide multilingual support and translation features.

Authentication:

- User authentication using technologies like JSON Web Tokens (JWT) for secure user access.

Responsive Design:

- Ensuring that the website is responsive and accessible across various devices and screen sizes.

Users profile

(Identify who will be using the system, in what mode, and their profile in terms of familiarity with using computers and such software.)

Who will be using the System: The General Public from a diverse range of regions, religions, and cultures from all over the globe, comprising of adults, teens and old people. Any individual with the internet and enthusiasm to learn more about cultures around them.

In what mode: User mode

Familiarity with computers: wide range from unfamiliar to too familiar however most of them would fall under the above average category

Feature requirements (described using use cases)

Read the instructions below and fill in the table. Delete all the blue text turning it in.

(This is a numbered list of use cases that are the features of the system to be implemented. Each use case is an operation that the user can perform on/with the system. For each use case, provide a description (2-3 sentences) so you know what to build and so you can write a test case to demonstrate that your system provides that feature. For each use case, you will identify (during release planning) the release in which it will be implemented: R1 or R2. Typically, your project will have 10-15 use cases, but feel free to add or delete table rows if you decide to use finer-grain or coarse-grain use cases).

No.	User Case Name	Description	Release
1.			
2.			
3.			
4.			
5.			
6.			
7.			
8.			
9.			
10.			
11.			
12.			
13.			
14.			

Use case diagram

Read the instructions below and fill in the table. Delete all the blue text before adding this to your repository or turning it in to your instructor.

Draw the UML use case diagram for the system. Make sure the use cases shown in the diagram correspond to the use cases described in the previous section.

Use case description

Delete all the blue text and fill-in the template before adding this to your repository or turning it in to your instructor.

Use Case Number:	UC-XX (Replace XX with a number)
Use Case Name:	Enter the name of Use Case
Overview:	Describe the purpose of the Use Case and give a 1-2 line description. This could be the same as the description provided in feature requirements section.
Actors:	List all actors that participate in this Use Case.
Pre condition:	Enter the condition that must be true before the main flow is executed.
Flow:	Main (success) Flow: Steps should be numbered.
	Alternate Flows: Include the post condition for each alternate flow if different from the main flow.
Post Condition:	Enter the condition that must be true when the main flow is completed.