**Audit Bazaar**

**Backend Technical Documentation**

## **Introduction**

This document provides an overview and technical details for the backend service of Audit Bazaar platform. The backend service is responsible for handling most of the functionality of the platform such as user activities, audits creation, etc.

## **Table of Contents**

* Technologies Used
* API Endpoints
* Middlewares
* Database
* Swagger Documentation
* Test Cases
* Dockerfile

### 1. Technologies Used

The backend service of Audit Bazaar utilizes the following technologies:

* Node.js: A JavaScript runtime environment.
* Express: A web application framework for Node.js, used to build APIs and handle middleware functionalities.
* MongoDB: A NoSQL database used for data storage.
* Swagger: A tool to generate API documentation and interact with APIs through a user-friendly interface.
* Jest and Supertest: Testing frameworks for creating and running test cases.
* Docker: Used for containerization and deployment.

### 2. API Endpoints

The backend service provides the following API endpoints:

* /register: Endpoints related to user onboarding.
* /login: Endpoints related login after onboarding.
* /getUserInfo: Endpoints related to user info management.
* /updateProfile: Endpoints related to user profile updation.

Detailed documentation for each API endpoint, including request parameters, response formats, and authentication requirements, can be found in the Swagger documentation .

### 3. Middlewares

The backend service utilises the following middlewares:

* Body Parser: Responsible for parsing incoming request bodies in JSON format, making the data accessible in the request object.
* CORS: Enables Cross-Origin Resource Sharing to handle requests from different domains.
* Dotenv: Loads environment variables from a .env file, facilitating configuration management.
* Jwt Validation: Involves a JWT validation check after login ensuring restricted usage of platforms for genuine users.

### 4. Database

The backend service uses MongoDB, a popular NoSQL database, to store and manage data related to the Audit Bazaar platform. MongoDB was chosen for its flexibility and scalability, which aligns with the platform's requirements.

### 5. Swagger Documentation

The backend service is equipped with Swagger documentation, which provides a user-friendly interface to interact with the APIs. It includes detailed information about each API endpoint, such as available methods, request parameters, and response formats. Developers can access the Swagger documentation at /swagger endpoint.

### 6. Test Cases

To ensure the reliability and correctness of the backend service, a comprehensive suite of test cases has been developed using Mocha and Chai testing frameworks. These test cases cover different functionalities of the service, including API endpoints and middleware functionalities. Regular testing is conducted to identify and fix any issues that may arise during development or maintenance.

### 7. Dockerfile

For easy deployment and distribution, the backend service includes a Dockerfile. The Dockerfile defines the environment and dependencies required to run the service, making it easy to deploy the application as a container on various platforms.

### 8. Conclusion

This technical documentation provides an overview of the backend service for the Audit Bazaar platform. It describes the technologies used, available API endpoints, implemented middlewares, database choice, Swagger documentation, test cases, and Dockerfile for containerization. Developers and stakeholders can refer to this documentation for a better understanding of the backend service and its functionalities.

## 