**Project**

**JSPR-Techies**

**Technical Document  
  
12/8/2023**

**Members**

Jayani Sumanka Gerine

Pallavi Dabade

Raksha Varahamurthy

### Surya Subramani

**Table of Contents**

Project Title & Members……………………………….…………….……. 1

Overview of the Software……………………………….…………….……………3

System Requirements………………………………………………………..…….5

Installation and Setup Instructions………………….…………….………………...6

User Guide……………………….……………………………..….………………...7

API Documentation……….………..……………….………………………………8

Architecture and Design Details…….……………….…………………………….9

Code Examples and Usage Scenarios……….…………………………………..10

Troubleshooting Guide…….……………….………………………………………11

Security Protocols……………………………………………………………………….12

Development and Deployment Procedures………………………………………….13

Maintenance and Support Information……………………………………………….14

**Overview of the Software**

Overview of the Software: A high-level description of what the software does, its purpose, and its intended users.

**Project Purpose:**

The purpose of the project is to develop a comprehensive platform that enhances the user experience for both students and professors within the university community. The platform leverages Artificial Intelligence (AI) algorithms to provide a personalized and efficient article search experience.

**Objectives:**

* User-Friendly Platform
* Informative Content
* AI assisted Search Recommender
* Collaboration and Community Building
* 24/7 Support and Continuous Improvement

**Main features:**

* Informative Content

JSPR-Techies contains well-researched articles written by students, professors and alumni of the University of Michigan academic community covering latest news and technological advancements in a wide variety of topics.

* Categorizing into departments and subtopics

Organizing content into categories gives an effective platform for readers to narrow down their search into what they are interested in.

* AI assisted Search Recommender

The AI assisted recommender is responsible for enhancing user experience and content discoverability. Users are recommended a list of top five articles based on the search keyword they enter.

* Collaboration and Knowledge Sharing Component

Facilitating collaboration and knowledge sharing is a fundamental aspect of our system.  It fosters a sense of community within the University of Michigan academic community.

* Customizable User profiles

The user can make a profile with information about the topics that interest him, and the system will make relevant article suggestions accordingly.

* User friendly interface design

While interacting with JSPR-Techies, we intend to provide the users a seamless and enjoyable experience by integrating interactive GIFs, logos, and vectors to our user interface design, making the process visually appealing and user-friendly.

**System Requirements**

System Requirements: Details about the hardware and software requirements needed to run the software effectively.

**Hardware Requirements:**

* **Server:**
  + Processor: Multi-core processor (e.g., Quad-core or higher)
  + RAM: 8 GB or higher
  + Storage: 100 GB SSD for blog data
  + Network: High-speed internet connection for efficient content delivery
* **Database Server:**
  + Database Type: MySQL
  + Processor: Dual-core or higher
  + RAM: 4 GB or higher
  + Storage: 50 GB SSD or higher for database storage
* **Load Balancer**:
  + Distributes incoming blog traffic across multiple servers to ensure high availability and performance during peak times.
  + Processor: Dual-core or higher
  + RAM: 4 GB or higher
  + Network: Gigabit Ethernet

**Software Requirements:**

* Operating System:
  + Server: Linux for optimal security and performance.
  + Content Delivery Network (CDN): Utilize a CDN service for efficient content distribution worldwide.
* Web Server:
  + Nginx or Apache: Configure to handle web requests and serve blog content efficiently.
  + SSL/TLS: Implement HTTPS for secure data transmission.
* Database Management System:
  + MySQL 8.x: Reliable and scalable database systems.
  + Database Optimization: Regularly optimize and index the database for performance.
* Blog Software and Plugins:
  + Latest Version: Keep the blogging software and plugins up to date for security and feature enhancements.

**Installation and Setup Instructions**

Installation and Setup Instructions: Step-by-step guidance on how to install and configure the software.

Step-1: Install Python 3.x and MySQL database if not already available.

Step-2: Install MySQL workbench.

Step-3: Create a virtual environment to isolate dependencies. To do that add the below script to your command prompt and activate it.

*For windows: $ python -m venv venv venv\Script\activate*

*For macOS/linux: $ python3 -m venv venv source venv/bin/activate*

Step-4: Install Flask and Flask-SQLAlchemy

*$ pip install Flask Flask-SQLAlchemy*

Step-5: Install project dependencies.

*$ pip install -r requirements.txt*

Step-6: Create MySQL database named ‘university’. Also create MYSQL user and grant privileges.

*CREATE DATABASE university ;*

*CREATE USER 'your-username'@'localhost' IDENTIFIED BY 'your-password';*

*GRANT ALL PRIVILEGES ON university.\* TO 'your-username'@'localhost';*

Step-7: Create the following tables.

* students

*CREATE TABLE `students` (*

*`umich\_id` int NOT NULL AUTO\_INCREMENT,*

*`first\_name` varchar(50) DEFAULT NULL,*

*`last\_name` varchar(50) DEFAULT NULL,*

*`major` varchar(50) DEFAULT NULL,*

*PRIMARY KEY (`umich\_id`)*

*) ENGINE=InnoDB AUTO\_INCREMENT=98989895 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci*

* articles

*CREATE TABLE `articles` (*

*`article\_id` int NOT NULL AUTO\_INCREMENT,*

*`title` varchar(1000) DEFAULT NULL,*

*`author` varchar(100) DEFAULT NULL,*

*`content` text,*

*PRIMARY KEY (`article\_id`)*

*) ENGINE=InnoDB AUTO\_INCREMENT=5655 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci*

Step-8: Configure the flask application and run the flask application.

*$ python app.py*

The application can be accessed on the URL: *http://127.0.0.1:5000.*

**User Guide**

User Guide: Instructions on how to use the software, including its features and functionalities.

1. The user clicks on the click button provided at the welcome wolverine page. Then the user enters the UMID. If the UMID exists in the system, the user is logged into the JSPR-Techies system. (add screenshot)
2. Next the user is brought to the page with the department heading to choose from. (add screenshot)
3. Further the user clicks on any one of the department headings as per his/her interest. (add screenshot)
4. This takes him to the next page asking him to enter keyword to narrow down his search. (add screenshot)
5. Further according to the entered keyword, the AI recommender shortlists top five articles and recommends it by displaying it to the user. (add screenshot)
6. The user clicks on the article that interests him from the list. (add screenshot)
7. The selected article loads on the page. (add screenshot)
8. The user is redirected to the list of articles where he can either select another article or he can logout. (add screenshot)

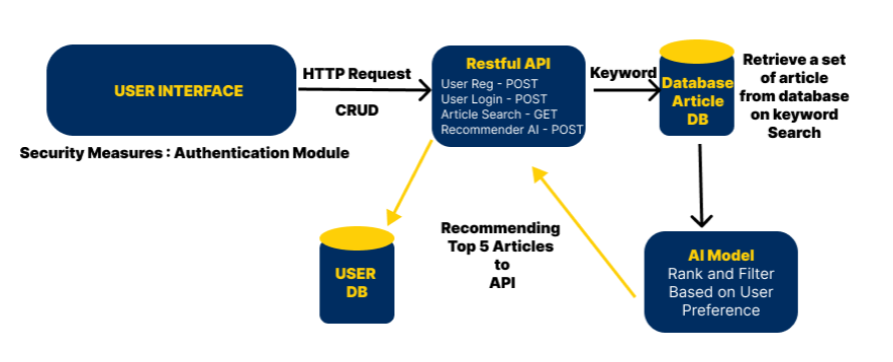
**API Documentation**

API Documentation (if applicable): Information for developers on how to interact with the software's APIs, including endpoints, request/response formats, and examples.

**Architecture and Design Details**

Architecture and Design Details: Information on the software's architecture, including data flow diagrams, architectural components, and design rationales.

**Context Diagram:**



**Glossary**

| **Words** | **Context** |
| --- | --- |
| umich | University of Michigan |
| User DB | Database to store user information |
| AI model | Ranking & filtering model for articles |
| API | Application Program Interface |

**Code Examples and Usage Scenarios**

Code Examples and Usage Scenarios: Practical examples of how to use the software or its components, are often useful for developers.

**Troubleshooting Guide**

Troubleshooting Guide: Common issues or problems users might encounter and their solutions.

**Login Issues:**

**Issue:** Users are unable to log in to their accounts.

**Solution:**

* Verify that the login credentials are correct and that the user account exists.
* Ensure CAPTCHA is user-friendly and not causing accessibility issues.

**Compatibility Issues:**

**Issue:** The blog doesn't display correctly on certain browsers or devices.

**Solution:**

* Ensure the blog theme is responsive and compatible with major browsers.
* Test the blog on various devices and screen sizes during development.
* Encourage users to update their browsers for optimal performance.

### **Slow Page Loading:**

**Issue:** Users experience slow loading times for blog pages.

**Solution**:

* Optimize images and multimedia content for the web.
* Implement browser caching to reduce load times for returning visitors.
* Consider using a Content Delivery Network (CDN) for faster content delivery.

### **Broken Links:**

**Issue:** Users encounter broken links when navigating through the blog.

**Solution:**

* Regularly perform link checks and update or remove broken links.
* Use proper redirection for outdated URLs.
* Implement a custom 404 error page with helpful navigation links.

**Security Protocols**

Security Protocols: Information on the security measures and protocols implemented within the software.

The security of the JSPR-Techies blog is crucial to protect sensitive information, maintain user trust, and prevent potential cyber threats. Below are security measures and protocols that should be implemented within the software of the University Tech blog:

### **User Authentication and Authorization:**

* **Description:** Use strong authentication mechanisms, such as multi-factor authentication (MFA).
* **Purpose:** Ensure that only authorized individuals have access to sensitive areas of the blog, reducing the risk of unauthorized access.

### **Password Policies:**

* **Description:** Enforce robust password policies, including complexity requirements and regular password updates.
* **Purpose:** Strengthen user account security and mitigate the risk of unauthorized access due to weak or compromised passwords.

### **Security Audits and Penetration Testing:**

* **Description:** Conduct regular security audits and penetration testing to identify and address vulnerabilities.
* **Purpose:** Proactively identify and fix security weaknesses before they can be exploited by attackers.

### **Backup and Recovery Procedures:**

* **Description:** Implement regular backup procedures for the blog's database and content.
* **Purpose:** Facilitate quick recovery in the event of data loss, accidental deletions, or security incidents.

### **Monitoring and Logging:**

* **Description:** Set up monitoring tools and log systems to track and analyze user activities, system events, and security incidents.
* **Purpose:** Detect and respond to security threats in real-time, allowing for timely intervention and mitigation.

**Development and Deployment Procedures**

Guidelines for developers on how to contribute to the software, including coding standards, version control practices, and deployment processes.

### **Development Procedures:**

#### **Version Control:**

* Use Git as the version control system.
* Create feature branches for each new feature or bug fix.
* Commit frequently with meaningful commit messages.
* Pull requests should be used for code review and merging changes into the main branch.

#### **Coding Standards:**

* Follow the PEP 8 style guide for Python.
* Use comments to explain complex sections of code.
* Adhere to consistent naming conventions for variables, functions, and classes.
* Ensure code readability and maintainability.

### **Deployment Procedures:**

#### **Environment Configuration:**

* Use environment variables to manage configuration settings.
* Clearly document configuration requirements for development, testing, and production environments.

#### **Continuous Integration:**

* Set up a CI/CD pipeline to automate testing and deployment processes.
* Run automated tests on each commit to the version control system.

#### **Database Deployment:**

* Use migration scripts to manage database schema changes.
* Backup the database before applying migrations in production.

#### **Deployment Strategies:**

* Implement blue-green or canary deployment strategies for minimizing downtime.
* Rollback procedures should be documented and tested.

#### **Post-Deployment Checklist:**

* Conduct post-deployment checks to ensure the blog is functioning as expected.
* Monitor for any performance issues or errors after deployment.

#### . **Monitoring and Logging:**

* Integrate monitoring tools to track the health and performance of the production environment.