**PRINTER SUPPORT SERVICE**

**Functional Components :-**

**Login :**

1. Authentication of users need to done whenever the user login

2. If a user enters the wrong password three times then he/ she needs to verify its

authenticity using OTP and only after that he/ she will be allowed to login again

within 24 hours and if the user login after 24 hours then OTP is not required.

3.User can reset the password if he/ she forgets the password, for it he/ she needs to enter the email/ mobile no. registered with and then verify the OTP.

**Register :**

1. For the first time when a user registers it needs to fill in necessary details such as email,

username, password, mobile number, etc.

2. When the user enters the necessary details after it, an OTP should be sent to its email or mobile number and that should be verified before proceeding further.

1. **Functionality: User table**

Store user information such as name, email, username, password, contact number, and address.

Allow users to register and create an account.

Provide user authentication and authorization.

Allow users to update their profile information.

Enable password reset functionality.

Track user activity and login history.

Support user roles and permissions.

**Non-Functionality:**

**Security:** Implement proper encryption and hashing techniques to protect user passwords.

**Performance:** Ensure efficient retrieval and storage of user information.

**Scalability:** Design the user table to handle a large number of users.

**Usability:** Create a user-friendly interface for user registration and profile management.

**Accessibility:** Ensure that the user table is accessible to individuals with disabilities.

**Product Table:**

**Functionality:**

Store product information such as product name, description, model number,product serial number.

Allow users to view and search for products.

Provide detailed product enquiry and features.

Enable product categorization and filtering.

Manage product inventory and stock levels.

Support product reviews and ratings.

Allow users to add products to their cart or wish list.

**Non-Functionality:**

Performance: Optimise product search and retrieval operations for quick response times.

Data Integrity: Implement data validation to ensure accurate and consistent product information.

Scalability: Design the product table to handle a large number of products.

Security: Protect product data from unauthorised access or modifications.

Usability: Design a user-friendly interface for product browsing and selection.

**Enquiry Table:**

**Functionality:**

Store user enquiries or support requests related to printer issues.

Capture details such as user name, email, enquiry description, date/time, and priority.

Assign a unique identifier to each enquiry for tracking purposes.

Enable status tracking (open, in progress, resolved) for each enquiry.

Provide a communication channel between users and support staff.

**Non-Functionality:**

**Performance:** Ensure efficient storage and retrieval of enquiry data.

Data Integrity: Implement data validation to ensure accurate and consistent enquiry information.

**Security:** Protect enquiry data from unauthorised access or modifications.

**Scalability**: Design the enquiry table to handle a large number of enquiries.

**Auditability:** Track and log changes made to the enquiry records for accountability purposes.

**Note:** The functionality and non-functionality mentioned above are just examples and should be adapted to the specific requirements and scope of your printer support service online project.

this is according to project

here some functionality are not in our project like payment and chat functionality don't include it

**Functionality:**

**User Registration and Authentication:**

Allow users to create an account by providing their name, email, username, password, contact information, and address.

Implement user authentication mechanisms such as email verification and password hashing for secure login.

Provide a login functionality for registered users to access their accounts.

Printer Support Request Management:

Provide a form or interface for users to submit printer support requests.

Capture details such as user information (such as name, contact information), printer model, description of the issue, and priority level.

Assign a unique identifier (SupportRequestID) to each support request for tracking purposes.

Store the date and time (Timestamp) when the support request is submitted.

Store the status of the support request (e.g., open, in progress, resolved).

**Support Ticket Tracking:**

Enable users to view the status of their support requests.

Provide updates and notifications to users regarding the progress of their support tickets.

Allow support staff to update ticket status and provide comments or solutions.

Implement a mechanism to track the history of status changes and user interactions for each support ticket.

**Knowledge Base and FAQ:**

Create a repository of printer-related knowledge articles, troubleshooting guides, and frequently asked questions (FAQs).

Categorise the knowledge base articles for easy navigation and retrieval.

Allow users to search for relevant articles based on keywords or specific printer models.

Provide a user-friendly interface to display the knowledge base articles with clear instructions and solutions.

**Live Chat or Messaging:**

Implement a real-time chat or messaging feature to facilitate direct communication between users and support staff.

Enable users to initiate a chat session with support agents.

Provide a messaging interface for users and support staff to exchange messages.

Support multiple chat sessions concurrently.

Store the chat messages in the database, including the sender, receiver, content, and timestamp.

**Payment Processing:**

Integrate a payment gateway to handle online payments for services or products related to printer support.

Implement secure payment processing to handle transactions.

Store transaction details, including the user, payment amount, timestamp, and payment status.

**Non-Functionality:**

**Security:**

Implement robust security measures to protect user data, including encryption and secure data storage practices.

Ensure secure communication channels for sensitive information, such as passwords and payment details.

Implement mechanisms to prevent unauthorised access to user accounts and support ticket information.

Performance:

Optimise database queries and data retrieval to provide fast response times.

Implement caching mechanisms to reduce server load and improve overall system performance.

Use indexing and query optimization techniques to improve database performance.

Scalability:

Design the database schema and system architecture to handle a growing number of users, support requests, knowledge base articles, and chat messages.

Consider scalability options such as sharding, replication, or cloud-based solutions to accommodate increased user traffic and data volume.

**Usability:**

Create a user-friendly and intuitive interface for easy navigation and interaction with the online printer support services.

Provide clear instructions and guidance to users for submitting support requests, accessing knowledge resources, and using the chat or messaging feature.

Implement responsive design to ensure compatibility and usability across different devices and screen sizes.

**Data Backup and Recovery:**

Implement regular database backups to ensure data integrity and provide mechanisms for data recovery in case of system failures or data loss.

Implement disaster

**Non-Functional Components :**

1. We will implement authentication and authorization using JSON Web Token (JWT).

2. Front-end will be implemented using React JS 18.

3. For styling we will be using Bootstrap v.5.2.3 Data Backup and Recovery:

4. Back-end will be implemented using ASP.Net using MVC Architecture.

5. For maintaining the database we will be using MySQL v.8.0.33

6. For continuation development we will use github.

7. Testing: a. Unit testing will be NUnit. b. Integration testing will be done using .NET Core Web SDK.

8. For deployment of our application on cloud we will be using an AWS EC2 instance