**S428 Spring 2025 Final Project:**

**Part A – The local website**

**Example:**

**"Mahatma Johndi’s Blog"** to dynamically post blog posts for users to view. This will allow the website contributors to post, edit, and update blog posts, and blog categories. It will be used to post a variety of different types of posts to express myself and share content to others.

**Key Features:**

* **CRUD Blog Posts & Categories**: Students can list textbooks and educational resources for lending or exchange. Users can reserve books, track borrowing periods, and search resources by title, author, subject, course code, or ISBN.
* **View, Like, Comment on posts:** Users can leave ratings and reviews about the condition of books and reliability of lenders. User profiles display reputation scores based on reviews and transaction histories. Right now, any user can like/dislike unlimited amount of times
* **Log in system**: User authentication that differentiates contributors from viewers, allowing contributors to access CRUD capabilities, and allowing viewers to view, like, and comment.
* **Messaging System**: A messaging interface for users to reach out directly to me

**Technology Stack:**

* **Backend**: Python (Flask)
* **Frontend**: HTML, CSS, JavaScript
* **Database**: SQLite

**Goal**: Enable students to apply full-stack development principles practically by addressing a real-world issue. The project encourages creative problem-solving, collaborative teamwork, and technical skill development.

**Project Requirements:**

**Backend Development**

* **Database Design & Management**:
  + User profiles (name, email, password, profile image, and location).
  + Blog Post Categories (categoryID, postID, more to come)
  + Blog Posts (postID, text content, images, likes, etc)
  + Comments (commentID, userID, postID)
  + Reviews and ratings for users.
* **Server-side Logic**:
  + User registration, login, and authentication (use Flask-Login).
  + CRUD operations for blog posts and categories
  + Search functionality based on blog category or title
  + Comment system on posts
  + Logic for ratings and reviews

**Frontend Development**

* **Responsive Web Design**:
  + Homepage showcasing recent listings and top-rated posts
  + User dashboard to view blog categories, recent posts, most popular
  + Blog pages
  + User profile view and edit page.
  + Comments under each post
* **Interactivity**:
  + Form validation and feedback (client-side and server-side).

**Security and Input Validation**

* Data validation and protection against SQL injections.
* Prevent issues from special characters (e.g., ;, ', ").
* Proper user authentication and session management.

**Appendix A: Database Description**

This project will use an SQLite database to store and manage data related to users, resources, messages, and reviews.

**Users Table**

* user\_id (INTEGER, PRIMARY KEY, AUTOINCREMENT)
* name (TEXT, NOT NULL)
* email (TEXT, NOT NULL, UNIQUE)
* password (TEXT, NOT NULL)
* profile\_image (IMAGE)
* username (TEXT, NOT NULL)

**Blog Posts Table**

* post\_id (INTEGER, PRIMARY KEY, AUTOINCREMENT)
* user\_id (INTEGER, FOREIGN KEY REFERENCES Users(user\_id))
* title (TEXT, NOT NULL)
* date\_posted (AUTMATIC DATE)
* body (TEXT, NOT NULL)
* likes ( INTEGER, DEFAULT 0)
* dislikes ( INTEGER, DEFAULT 0)
* image\_path (TEXT)

**Comments Table**

* comment\_id (INTEGER, PRIMARY KEY, AUTOINCREMENT)
* user\_id (INTEGER, FOREIGN KEY REFERENCES Users(user\_id))
* post\_id (INTEGER, FOREIGN KEY REFERENCES Blog Posts(post\_id))
* timestamp (Automatic date, not null)