Blog Web Application

General Description:

The proposed web application allows users to create and share blog posts. The user can:

1. Display all blog posts sorted by date (with the most recent posts displayed first).

2. Display a list of all authors.

3. Filter and display only the blog posts composed by a selected author.

4. Select a specific date from a calendar and display all posts written on that day.

5. Register in the system with a unique username and password and start writing their own blog posts.

6. (Optional) Search and find all blog posts that match a specific user query.

Notes:

1. Each blog post is associated with an author and consists of a title, text content, and creation date.

2. Initially, the database should contain approximately 5 authors and about 20 blog posts.

3. The system should not display more than a certain number (P) of posts per page. If there are more matching posts in the system, the app should display "Next" and "Previous" links for navigating through the pages of results.

4. For the search function, users can type a specific string in the search box, and the system will return all blog posts with that string in the title.

Proposed Architecture:

The system will be implemented using a client-server architecture. We will use the Django framework for building the main application logic.

Database Structure:

The database will contain three tables: authors, blogposts, and users. The structure will be as follows:

Table authors

- id: int (unique author ID)

- name: char(20) (author name)

Table blogposts

- id: int (unique blog post ID)

- title: char(50) (blog post title)

- content: text (blog post content)

- date: date (creation date)

- author\_id: int (ID of the author - foreign key)

Table users

- id: int (unique user ID)

- username: char(20) (username)

- password: char(20) (password)

Project Plan:

1. Set up the system (install all required software tools).

2. Create the database and design its structure.

3. Create the business logic (ensure it works without a user interface).

4. Create a prototype of the user interface in plain HTML.