# BLOG POST OOP PROJECT.

Absolutely! Here's a project idea that does not require JavaScript for the creation part:

\*\*Simple Blog Management System\*\*

Create a basic blog management system where users can register, log in, create blog posts, and view posts by other users. This project will still allow you to work with Servlets, Threads, and basic design patterns while avoiding the need for complex client-side JavaScript.

1. \*\*User Registration and Login:\*\*

Implement Servlets for user registration and login functionality. Use a database to store user credentials securely.

2. \*\*Blog Post Creation:\*\*

Build Servlets to handle the creation of blog posts. Users should be able to write, submit, and store their blog posts in the database.

3. \*\*Blog Post Viewing:\*\*

Create a Servlet to display all blog posts on the homepage. Users can see their own posts and posts from other users.

4. \*\*Simple Design Patterns:\*\*

Incorporate basic design patterns like Singleton for managing the database connection and perhaps Observer for handling real-time updates (if needed).

5. \*\*Thread Management:\*\*

Use Threads to handle concurrent access to the blog posts and ensure data integrity when multiple users are accessing the application simultaneously.

6. \*\*Web Interface:\*\*

Create a straightforward and user-friendly web interface using HTML and CSS. The interface should display the blog posts, login/register forms, and allow users to interact with the application.

7. \*\*Security Considerations:\*\*

Ensure that user data is secure by implementing HTTPS for data transmission and storing passwords securely in the database.

This project will focus on the essential functionality of a blog management system and can be completed without the need for JavaScript. It will provide you with a good understanding of working with Servlets, Threads, and basic design patterns while building a useful web application. Happy coding!

DIVISIONS.

Certainly! Let's divide the work for four people to build the Simple Blog Management System:

1. \*\*ARNOLD: Front-End Developer\*\*

Responsibilities:

- Create the user-friendly web interface using HTML and CSS.

- Design and implement the homepage to display blog posts.

- Design and implement the login and registration forms.

- Ensure that the web interface is responsive and works well on different devices.

2. \*\*CONSLATA: Servlet Developer\*\*

Responsibilities:

- Implement Servlets for user registration and login functionality.

- Build Servlets to handle blog post creation and submission.

- Develop the logic to store blog posts in the database.

- Handle authentication for user access to create and view blog posts.

3. \*\*TIFFANY: Database Developer\*\*

Responsibilities:

- Set up the database to store user credentials and blog posts.

- Design the database schema for users and blog posts.

- Implement database interactions using JDBC or an ORM (Object-Relational Mapping) framework.

- Ensure data security and handle data storage and retrieval efficiently.

4. \*\*ANN: Thread Management and Design Patterns\*\*

Responsibilities:

- Manage concurrent access to the blog posts using Threads.

- Implement Thread synchronization to ensure data integrity.

- Incorporate basic design patterns like Singleton for managing the database connection.

- If real-time updates are required, use the Observer pattern to handle them.

Throughout the development process, the team should collaborate on integrating their respective components and ensure the proper functioning of the blog management system. Regular communication and code reviews will help identify potential issues early and maintain the overall quality of the application.

By dividing the work among team members, you can efficiently build the Simple Blog Management System, and each person can focus on their specific areas of expertise. Happy coding!