Project Requirements Document(PRD)

Title: Blog App

# Overview:

Build a Django-based blogging platform where users can register, log in, create, edit, and delete posts with rich-text formatting using Quill.js.  
The project helps you learn user authentication, CRUD operations, and front-end integration with HTML, CSS, and JavaScript.

# Goals & Objectives:

* **Create a blog web application using Django** that allows users to register, log in, and log out securely.
* **Enable users to create, read, update, and delete posts**, with each post linked to its author.
* **Use Quill.js for rich text editing** and Bootstrap for a responsive, user-friendly interface.
* **Display all posts on the home page** and allow users to view only their posts in a separate section.

# Features

| **Feature** | **Description** | **Priority** |
| --- | --- | --- |
| View Posts | Display a list of all user blog posts with author and date | High |
| Create Post | Authenticated users can write posts using Quill rich editor | High |
| My Posts | Users can view and manage only their own posts | Medium |
| Delete Post | Users can delete their own posts from the platform | High |
| Post Details | Show full post content with formatted HTML | High |

# Functional Requirements

# **View Events** Users can browse upcoming events, each displaying relevant information such as the event title, description, scheduled date and time, type (e.g., workshop, webinar), and location. All event data is stored and managed using SQLite for efficient and lightweight local storage.

# **Reminder System** A simple reminder feature allows users to receive local notifications or alerts for upcoming events or posts. Users have control over this feature and can choose to enable or disable reminders according to their preferences.

# **Filter System** To improve navigation, the platform includes filters based on event type, date, or other categories using dropdown menus or button selections. An optional search bar helps users quickly locate specific events or posts by keywords.

# UI/UX Requirements

# Clean, mobile-first responsive design for seamless viewing across devices.

# Three main user interface screens:

# **Home:** Displays a list of all blog posts or events.

# **Post Details:** Shows the full content of a selected post along with author and date.

# **Create Post / Reminder Settings:** Interface for users to create new posts or manage post-related reminders.

# **Tech Stack**

# **Frontend:** Built using HTML, Tailwind CSS, and basic JavaScript for dynamic interactions.

# **Backend:** Powered by Django with SQLite as the database for storing posts, users, and comments.

# Timeline

| Task | Deadline |
| --- | --- |
| Requirement Finalization | Day 1 |
| UI Wireframes | Day 2 |
| Build Event List View | Day 3 |
| Add Filter & Reminder | Day 4 |
| Testing | Day 5 |
| Final Touches & Report | Day 6 |

# Success Criteria

* App should load and display at least 5 posts.
* User can create a new post and optionally delete it.
* Mobile-responsive layout with clean UI.
* No major bugs found during testing.

# Deliverables

* Fully working blog app (deployed via PythonAnywhere or similar).
* Source code hosted on GitHub.
* Short demo video (2–3 minutes showing key features).
* Final project report (PDF including screenshots, features, and learnings).