Abstract: Task Manager Application

1. Introduction

The Task Manager is a visual tool designed to help small and medium-sized teams collaborate on shared projects. By combining an intuitive board-and-card interface with access controls and real-time interactions, the application assists with task planning, tracking, and completion. Users can quickly sign in, create or join boards, and organize work into lists and cards.

2. Content and Concept

2.1 Key Features

1. Access Management: Secure authentication (Google OAuth 2.0 + web tokens) ensures only authorized users can create or modify data. Boards offer three visibility levels (Private, Internal, Public) to match collaboration needs.
2. Board Organization: Each board serves as a top-level container. Boards include metadata (name, visibility, member list) and support dynamic ordering via drag-and-drop or manual repositioning.
3. List Management: Within boards, ordered lists represent workflow stages. Users can add, rename, reorder, copy, or delete lists with minimal effort.
4. Card Management: Cards encapsulate individual tasks and hold rich content: titles, descriptions, attachments, due dates, and member assignments. Comment threads with timestamps enable asynchronous discussions. Cards can be moved across lists to reflect progress.

2.2 Benefits

- Speed & Simplicity: Quick onboarding and minimal setup.
- Visual Clarity: Drag-and-drop interactions make task flows transparent.
- Collaboration Focus: Centralized communication within team.
- Scalability: Optimized for small to mid-sized teams.

3. Technical Approach

This section outlines the architecture, development processes, and testing strategies adopted during the project.

3.1 Architecture Overview

- Frontend: Built with Vue.js3 framework (Java Script) and Bulma CSS.
- Backend: Implemented with Django 5 framework, Python baed. Endpoints handle CRUD operations for users, boards, lists, and cards, with middleware for authentication and authorization.
- Database: Sqlite3 stores the data like users, boards, lists, cards, attachments, comments, etc. Schemas are normalized and indexed for efficient querying of popular operations (e.g., fetching a user's boards).
- Authentication is implemented wuth Google Oauth solution, web-tokens for authenticated users are stored in the database.

3.2 Security and Performance Considerations

- Authentication & Authorization: OAuth 2.0 integration followed best practices for token management and session security. Role-based access checks prevented unauthorized data access.
- Data Validation: front end validation sanitizes inputs and prevented malformed data from reaching the database (for example, only correct emails can be entered).

3.3 Documentation and Comments

- Documentation: Readme.md file contains installation instructions that are easy to follow.
- The source code include the comments for better understanding of classes and functions used.

4. Conclusion

The Task Manager Application balances ease of use with functionality, enabling teams to collaborate effectively. Through a well-defined architecture the project delivered a maintainable and scalable codebase.