

TaskTracker: Project Report

1. Introduction

This report documents the TaskTracker project built to satisfy the BuildYourOwnProject guidelines.

2. Problem Statement

Students and small teams need a lightweight task manager with analytics and simple prediction features.

3. Functional Requirements

User management, Task CRUD, Analytics, Export, Predictor.

4. Non-Functional Requirements

Non-Functional Requirements

- Performance: Responses under 300ms for common operations (demo)
- Usability: Simple and clear UI with minimal steps
- Reliability: SQLite transactions and basic error handling
- Maintainability: Modular code with clear separation of concerns
- Security: Demo-level session management (not production)
- Logging: Basic logging for errors and key events

5. System Architecture

Flask app (src/app.py) -> SQLAlchemy -> SQLite. ML model saved in models/

Use Case Diagram

Register

Create Task

Update Task

Complete Task

View Analytics

Export Report

Class Diagram

User

Task

TaskService

AnalyticsService

Predictor

Sequence Diagram: Create Task

User -> Web UI

Web UI -> TaskService

TaskService -> DB

DB -> TaskService -> UI

ER Diagram

User(id,username)

Task(id,title,difficulty,priority,completed,actual_time)