

TASKY

By Ethan Howlett, Corbin Coles, Kaiden Alva and Porter Jordan

Project Overview

+

•

◦

Purpose: Allows users to keep track of their tasks on a browser-based environment

Audience: Primarily College Students => Can include working adults

Main Problems Solved:

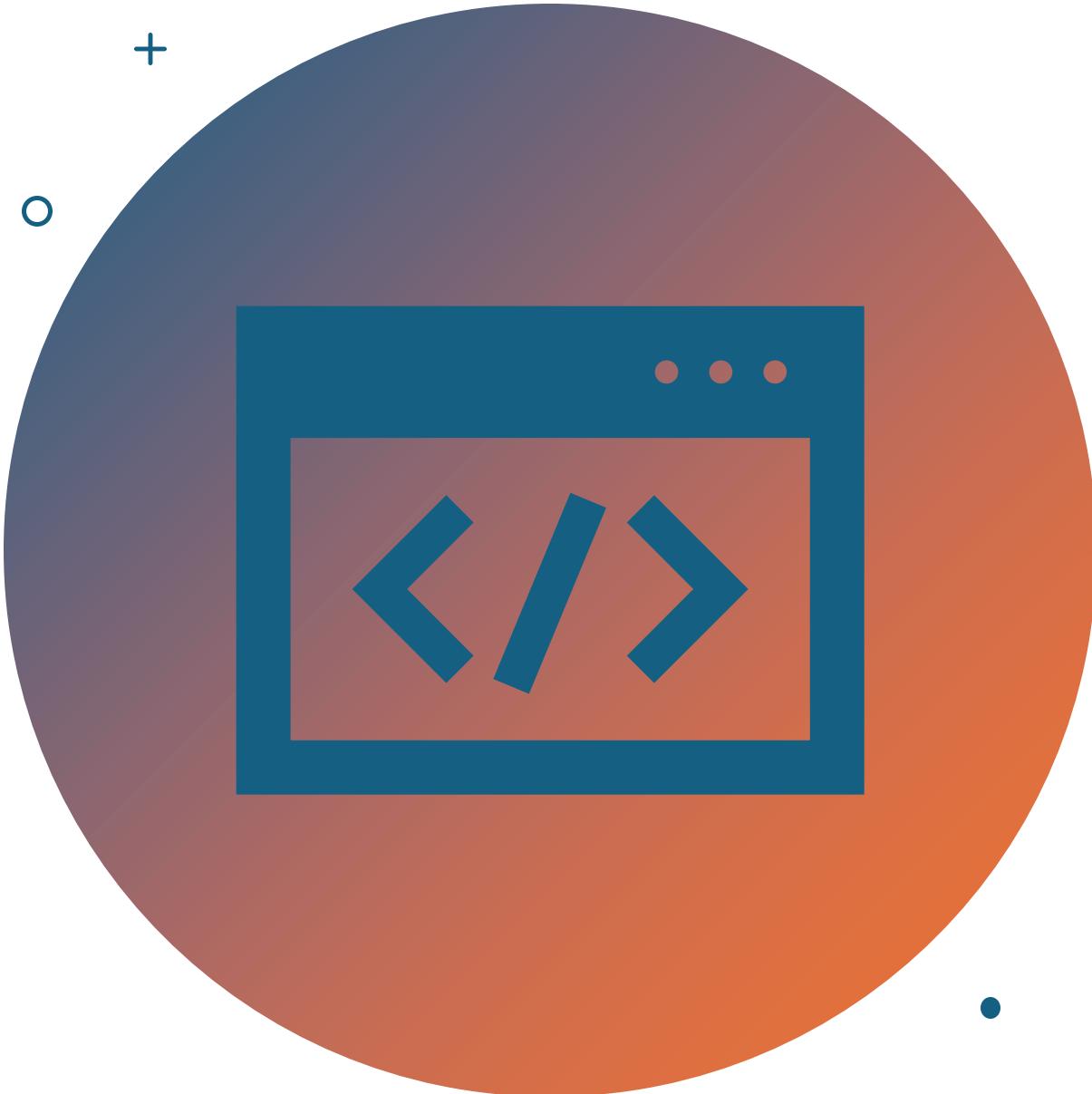
- **Forgetting deadlines:** Missing important due dates and appointments
- **Being overwhelmed by workload:** Difficulty prioritizing and managing multiple tasks
- **Disorganized notes and tasks:** Scattered information across multiple platforms

Development Process

- +
-
-

Requirements

- Task Manager
- User Authentication
- User Friendly Frontend
- Useful API
- Database



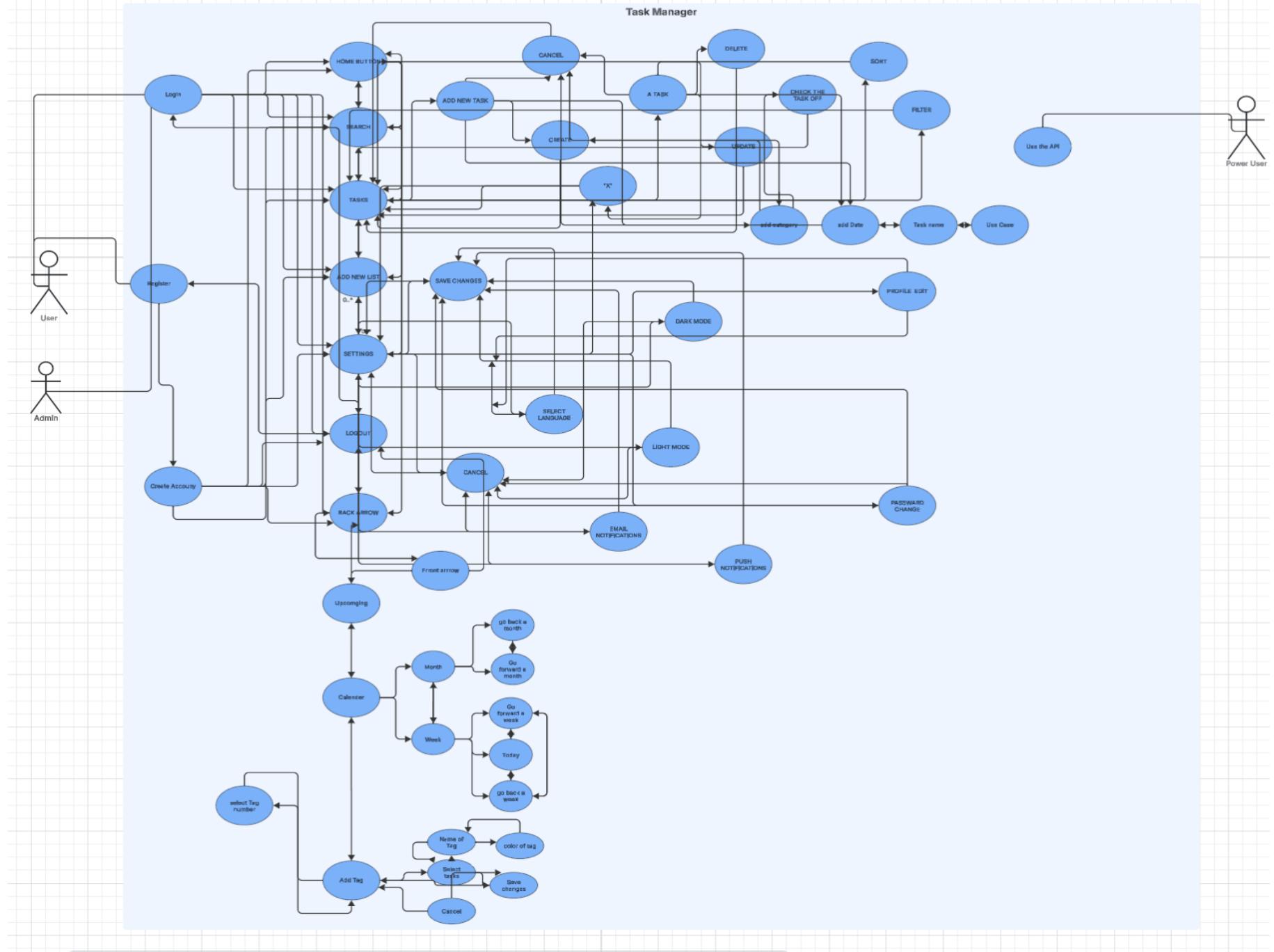
Outstanding Features

- Dedicated backend (Python) and frontend (React – TypeScript)
- Dedicated database (SQLite, easily migrate to Postgres)
- FastAPI for Api routes
- User Authentication and session key
- Visually appealing and easy to use

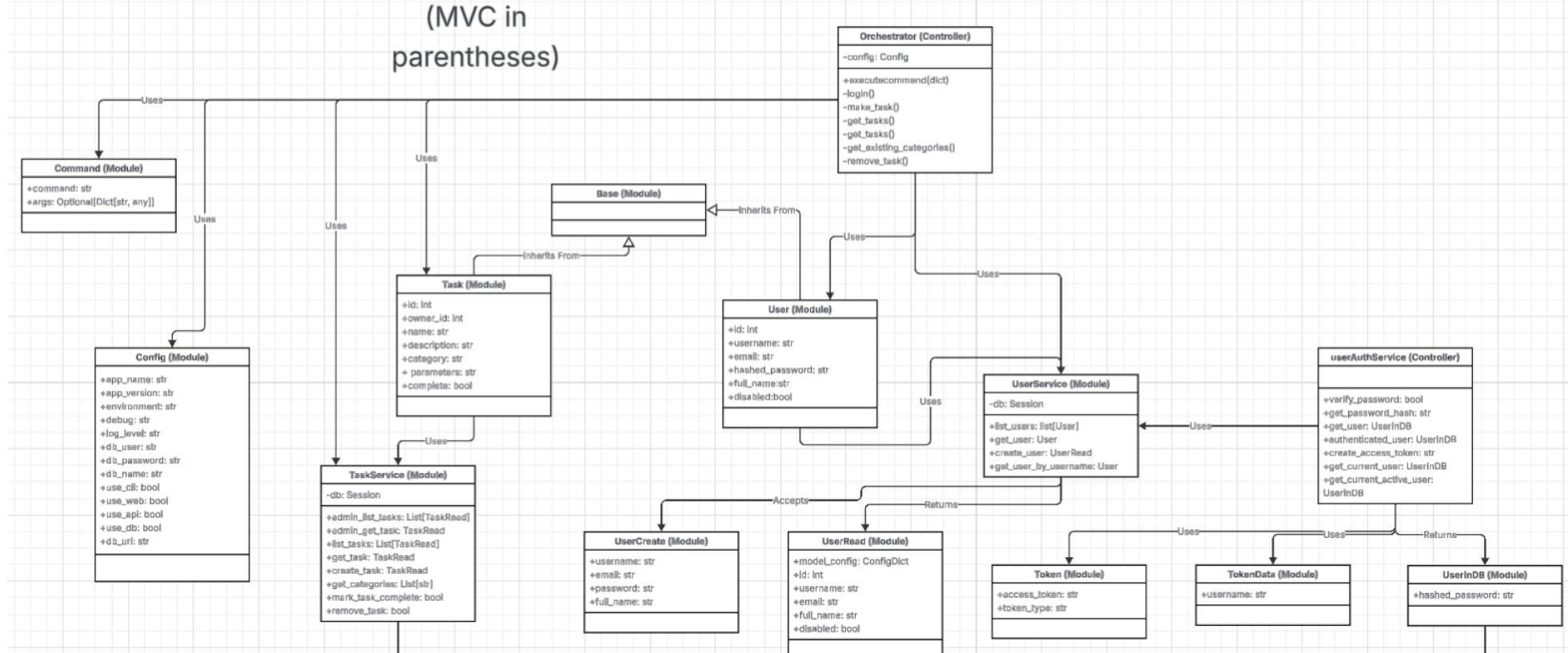
Backlog

Title	...	Type	...	Priority	↑	...	Assignees	...	Status	...
1 ⓘ Simple and Beautiful UI/UX: Intuitive and visually appealing user interface #15		Feature		High			Ethan-Howlett and...		Done	
2 ⓘ Enables users to delete tasks they don't need #4		Feature		High			Kaiden-Alva		Done	
3 ⓘ Persistent Data: Save users to-do-lists (tasks & categories) #18		Feature		High			CorbinColesA259 ...		Done	
4 ⓘ Task Management Core: The ability to add tasks & their information to a to-do list #6		Feature		High			CorbinColesA259 ...		Done	
5 ⓘ Create a Functional API #3		Feature		High			CorbinColesA259 ...		Done	
6 ⓘ Automatic Scheduling: Tasks can automatically be added to the user's schedule, helping them stay organized #13		Feature		High					Backlog	
7 ⓘ Reminder Settings: Adjustable amount of days for reminders #8		Feature		High					Backlog	
8 ⓘ Multiple Lists: Creating and managing multiple to-do lists (Categories) #10		Feature		High					Done	
9 ⓘ Calendar Integration: Integrate scheduled tasks around existing events and/or classes #12		Feature		High			Kaiden-Alva and p...		Done	
10 ⓘ Make task api #30		Task		High			Kaiden-Alva		Done	
11 ⓘ Enables users to change the app from light mode to dark mode #2		Feature		Medium			Ethan-Howlett and...		Done	
12 ⓘ Categorization & Prioritization: Categorize/tag reminders/todos and label priority levels #14		Feature		Medium			CorbinColesA259 ...		Done	
13 ⓘ Enables users to change the color or background image of their planner #16		Feature		Medium			Ethan-Howlett and...		Blocked	
14 ⓘ Automatic Planning Rules #17		Feature		Medium					Backlog	
15 ⓘ Task Suggestions #5		Feature		Low					Backlog	
16 ⓘ Push Notifications #7		Feature		Low					Backlog	
17 ⓘ List Merging #9		Feature		Low					Backlog	
18 ⓘ Delete tasks #22		Task					Kaiden-Alva		Done	
19 ⓘ Time Scheduling: Schedule times during each day to work on tasks during a week #11		Feature							Blocked	
20 ⓘ Mark tasks as complete #23		Task					Kaiden-Alva		Done	
21 ⓘ Dev #26							Ethan-Howlett and...		Done	
22 ⓘ Update pydantic models #32		Task					CorbinColesA259 ...		Done	
23 ⓘ Update class diagram #25							CorbinColesA259...		Done	
24 ⓘ Create new functions for new api's #33		Task					CorbinColesA259 ...		Done	
25 ⓘ Create login page #35		Task					Ethan-Howlett		Done	
26 ⓘ Merge dev into main #43									Done	
27 ⓘ Implement home/upcoming tasks page #46		Task					Kaiden-Alva		Done	
28 ⓘ Update database schema #31		Task					CorbinColesA259		Done	
29 ⓘ Test Task functions with DB, mostly unit testing #49		Task					Kaiden-Alva and p...		Done	
30 ⓘ Build tests for delete and marking tasks complete #24		Task					Kaiden-Alva		Done	
31 ⓘ Test User functions with the DB #50		Task					CorbinColesA259		Done	
32 ⓘ Create Integration tests for the APIs #51		Task					Ethan-Howlett		Done	

USE CASE DIAGRAM



CLASS DIAGRAM



Engineering Practices

Coverage report: 75%

Files **Functions** **Classes**

coverage.py v7.11.3, created at 2025-11-10 18:08 -0700

File ▲	statements	missing	excluded	coverage
backend/api/v1/task_api.py	62	14	0	77%
backend/api/v1/user.py	162	68	0	58%
backend/core/config.py	20	0	0	100%
backend/core/logging.py	3	0	0	100%
backend/db/init_db.py	39	11	0	72%
backend/db/schema.py	33	0	0	100%
backend/main.py	51	22	0	57%
backend/models/task.py	109	11	0	90%
backend/models/user.py	75	14	0	81%
backend/services/task_service.py	127	31	0	76%
backend/services/user_auth_service.py	91	44	0	52%
backend/services/user_service.py	194	70	0	64%
Total	966	285	0	70%

coverage.py v7.11.3, created at 2025-11-10 18:08 -0700

File ▲	statements	missing	excluded	coverage
total	966	285	0	70%
yd.services.lens.services.lens	461	10	0	94%
yd.services.lens.services.lens	87	44	0	23%
yd.services.lens.services.lens	153	70	0	55%

Design Patterns

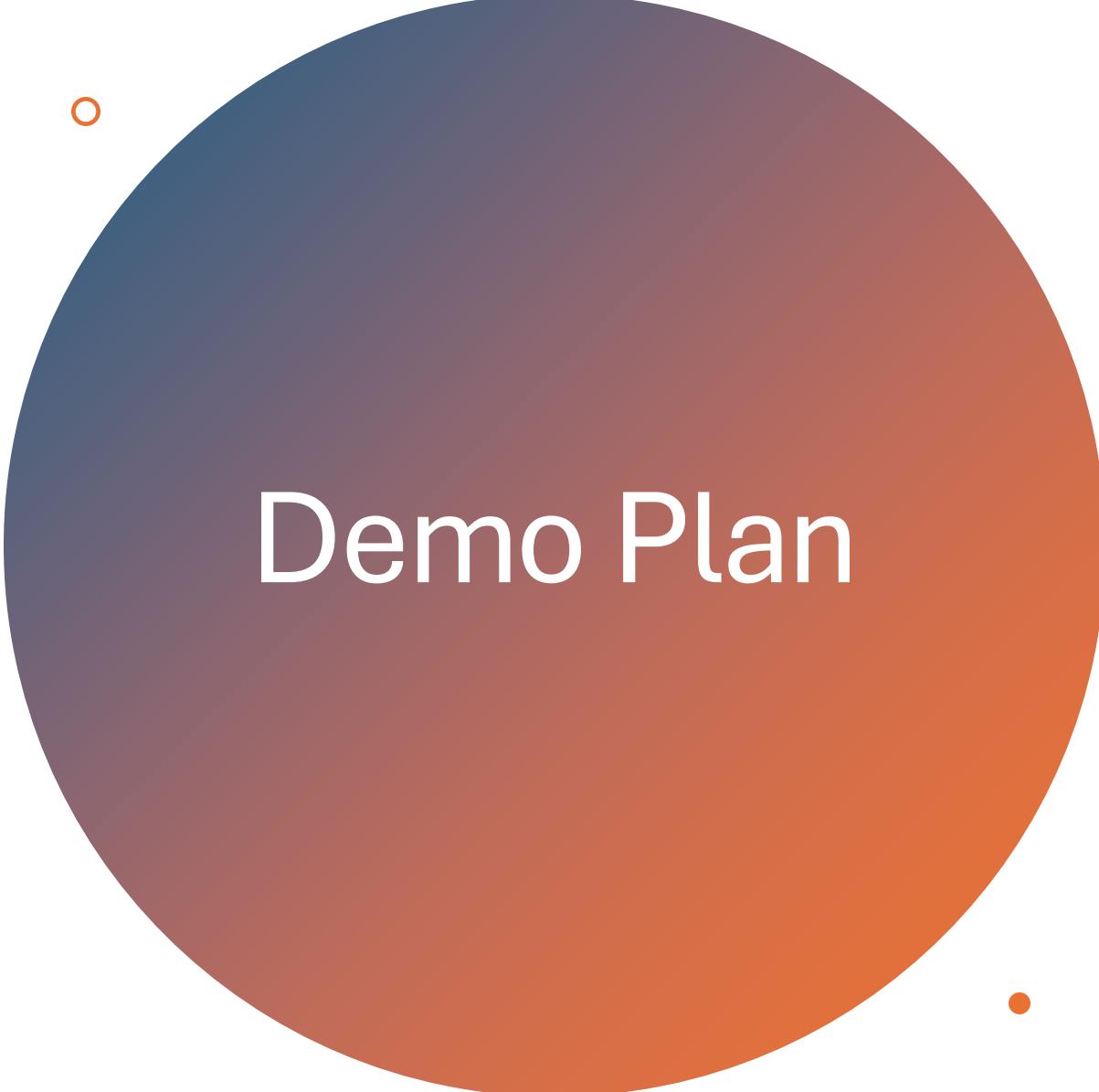
- Service Layer Pattern
- Singleton Pattern (Yielding services, Callender implementation)
- MVC (ModelView Control)
- Pydantic models and input validation

Design Principles

- CRUD (Create Retrieve Update Delete)
- SOLID

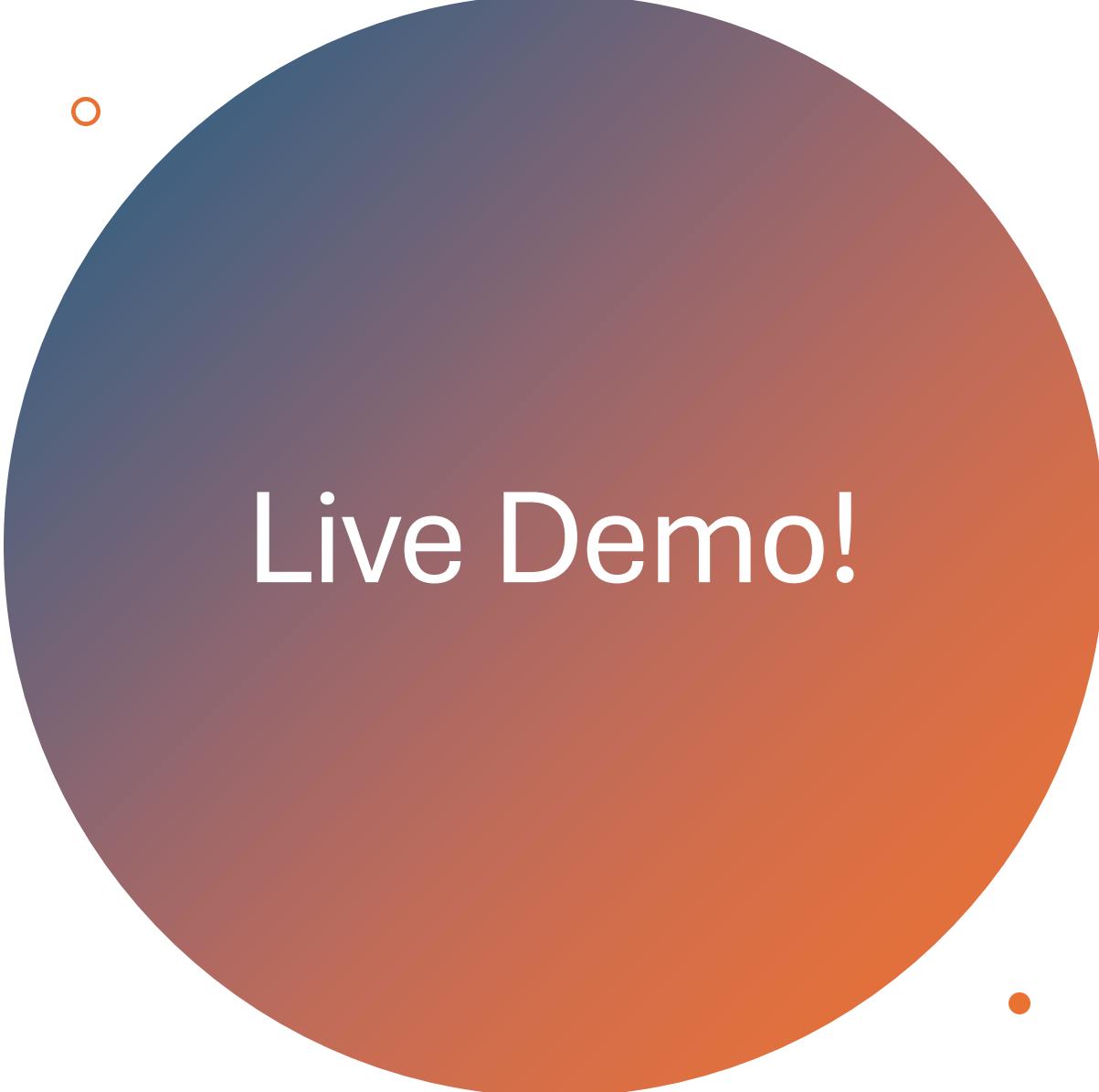
Testing

- Pytests / coverage
 - Unit Testing (Service class functions)
 - Integration Testing (testing through API)



Demo Plan

1. Register as a User
2. Create a task
3. Go to Calendar to see task
4. Make a Category
5. Create 2 tasks and put them under different categories
6. Edit a task
7. Mark a task complete
8. Delete a task
9. Refresh page or close the tab then re-open
10. Logout
11. Log back in



Live Demo!

Everyone go to
tasky.stingerdev.com

Future Enhancements

- Notifications
- Editing lists
- Tagging tasks
- AI and LLM integration

• Questions/Comments?

