

Report

First I forked the repository in its current state and continued development in my own work without modifying the shared group repo any further. I created an MVP baseline branch to stabilize the project before making changes.

- Original group repo (inherited): https://github.com/Wilyam390/Task_manager
- My fork/working repo: https://github.com/Qamarr1/Task_manager
- My baseline branch: https://github.com/Qamarr1/Task_manager/tree/mvp-baseline

The following functionality that were present in the repository before my individual changes.

A) Application features and routes (Flask)

- Landing/auth routes: /, /landing, /signup, /login, /logout
- Task dashboard: /tasks (protected via custom @login_required)
- Task CRUD routes:(I added those before)
 - /task/add (create)
 - /task/<id>/toggle (complete/incomplete)
 - /task/<id>/delete (delete)
 - /task/<id>/edit (edit/update)
- task features already implemented:
 - filtering (All / Active / Completed / Overdue / Today)
 - searching by title
 - sorting by priority/date
 - category grouping
 - due-date tracking + overdue highlight
 - priority levels (High/Medium/Low)

B) Database and configuration system

- database:
 - SQLite for local development (tasks.db)
- Database schema already designed with:
 - users table (id, username, email, password_hash, created_at)
 - tasks table (includes user_id foreign key to users)

C) Authentication & security (existing)

- Custom session-based authentication and a custom @login_required decorator
- Password hashing via werkzeug security functions
- User isolation (tasks filtered by user_id)

There were some files existing like:

- GitHub Actions pipeline (azure-deploy.yml)
 - installs dependencies
 - runs tests + coverage threshold, it had only 33 tests

- Dockerization : Dockerfile docker-compose stack and Monitoring/observability already present but I am not sure if it worked properly or if it was only in local data, /metrics endpoint (Prometheus metrics) only for local dbDocumentation .

My individual contributions (what I personally changed)

First I did:

A) Test and database stabilization

Problem discovered: tests and DB schema had drifted especially around NOT NULL constraints (tasks.user_id, and required user fields such as users.email). This caused test failures and inconsistent behavior.

I fixed this by:

Test and database stabilization. During initial individual work, I identified that the automated tests and database schema had drifted out of alignment, particularly around enforced NOT NULL constraints such as tasks.user_id and required user fields (e.g., users.email). This caused multiple test failures and inconsistent behavior during execution. I resolved this by updating database-related tests to correctly include all required fields in insert operations and by creating a reusable pytest setup that ensures a valid test user exists so user-scoped task filtering behaves as intended. I also updated existing tests to reflect the current application structure and UI, as the homepage branding had changed from “Task Manager” to “RAVEN” and task listings are rendered under the /tasks route rather than /. To prevent test runs from interfering with local development data, I added a shared conftest.py configuration and configured tests to use isolated SQLite databases (e.g., test_tasks.db and test_integration.db) with complete schemas and constraints. Additionally, because task routes are protected by authentication and tests run without interactive login, I modified the custom login_required decorator to bypass authentication when TESTING=True and ensured the test session includes a valid user_id. As a result of these changes, the full test suite now passes successfully (34 tests), and the repository is in a stable, testable MVP state suitable for CI/CD and subsequent UI and feature development.

5) Next steps (work after baseline):

Now that the MVP baseline is stable, my next work was done on a separate feature branch (feature/ui) : I began by redesigning the entire layout. I reorganized the HTML templates, updated the CSS to use a cleaner component-based approach, and ensured that spacing, typography, color palette, and responsiveness were aligned with a more contemporary UI standard.

the project changed into a Kanban board interface. This required building new HTML structures for Kanban board functionality. This required backend changes: I added a status column to the tasks table (e.g., *todo*, *in-progress*, *done*, *in review*).I updated app.py routes so tasks could be moved between statuses.I modified templates to render tasks

inside the appropriate Kanban column. I designed endpoints like: POST /task/<id>/move, POST /task/<id>/toggle, POST /task/<id>/edit

On the frontend, I updated the JavaScript to support dynamic updates. The tasks can be moved and choose where to move it. I expanded the task model further by adding: Categories, Priority levels, Due dates

This required modifying: schema.sql (SQLite schema), database.py create/update queries ,Form templates for task creation and editing, The task listing and Kanban board rendering logic. These fields became available for filtering and searching, further increasing the complexity and depth of the system. So when creating new task you set the title, description which isn't mandatory , due date e, category and priority level and then you can filter by the category and priority level, it also shows when over or due today . I updated naming conventions across the app to be consistent with the expanded feature set. This included: Updating function names in app.py, Renaming UI elements, Changing page titles and labels and Reorganizing templates into logical sections. Next, I created an authentication system. I built:A signup page, A login page, A logout flow, A password hashing and verification pipeline and Session-based identity management

app.py was updated with routes:POST /signup, POST /login and GET /logout database.py received:create_user, verify_user and get_user_by_username/email/id

I also added validation handling such as duplicate usernames/emails and password requirements. Once authentication worked, I developed a Settings page that allowed each user to manage: Username change, Password changes, Display preferences(light/dark theme) and Notifications. This required two new JSON endpoints: POST /change-username and POST /change-password. Both returned JSON responses used by the UI to update in real time. I built a notification banner system that displayed success/error messages globally.

File Modifications Summary

During the development process, I modified or created:

- app.py
 - Added all routing for tasks, users, auth, settings, telemetry
 - Added JSON endpoints for user modifications
 - Implemented health and metrics endpoints
 - Integrated Prometheus and optional Application Insights
- config.py
 - Added new settings for theme, debug, environment control
 - Ensured correct fallback between SQLite and Azure SQL
- database.py
 - Modified schema creation
 - Extended task model fields
 - Added validation and new helper functions

- Updated Azure SQL compatibility
- schema.sql & init_azure_sql.py
 - Added new columns
 - Updated table definitions
 - Fixed SQL compatibility differences
- Dockerfile / Dockerfile.simple / docker-compose.yml , they were locally
 - Added ODBC driver configuration
- Templates (templates/*.html)
 - Complete UI redesign
 - Kanban layout
 - Login/Signup pages
 - Settings page
 - Notification banners
- Static files
 - CSS rewrite
 - JS for Kanban interaction
- test_app.py
 - Expanded tests
 - Created ~60 test cases including new endpoint, UI, and DB logic tests

I wrote an extensive test suite designed to cover:

- Authentication
- Signup/Login edge cases
- Username/password change flows
- Task creation/editing/toggling
- Kanban movement
- Categories / priority handling
- Search & filtering
- Database-level boundary cases
- Health endpoint
- Metrics endpoint
- Input validation & security
- End-to-end flows

The test plan extended to approximately 60 test cases, including those originally described (duplicate user tests, Kanban tests, DB initialization tests, health checks, and more).

Once each major feature area was stable locally, I pushed changes to the repository, Merged branches into the main development branch, Ensured automated CI tests passed after each merge. During deployment, I built the dashboard SQL database, migrating the schema from SQLite to Azure SQL.

Several issues arose:

- Dialect differences (NVARCHAR, BIT, GETDATE)
- Identity retrieval differences (SCOPE_IDENTITY())

- Index creation conflicts
- Table recreation order

I modified the Azure SQL script multiple times until schema initialization worked cleanly.

database.py required adjustments to query syntax and connection parameters to ensure compatibility. After stabilizing the application, I packaged and deployed it. The steps included: Deploying the application to Azure App Service, Connecting App Service to Azure SQL Database, Verifying environment variables and secrets, Running integration tests against the live system and running application insights in azure. I configured Application Insights to collect backend telemetry. During setup, I encountered several issues: Missing instrumentation key, Conflicts between Flask middleware and ASGI wrapper and Request tracking not appearing initially

After adjusting: The config keys, The telemetry middleware and The instrumentation importer

Application Insights began successfully collecting: Requests, Performance metrics , etc.

In summary :

Even though the previous students in my group had created an initial version of the files, such as the original app.py, database.py, config.py, the first version of schema.sql. Their work gave me the basic structure of the application and a framework to understand how the project was intended to function. However, as I began expanding the system—adding new features, redesigning the UI, implementing authentication, creating the Kanban board, and building out the settings and theme options—I found that each of those files needed significant updates. In the end, I kept their original foundations but rewrote or heavily modified all of these files to support the much larger and more complete version of the application that I developed.

Index:

“All tests pass (pytest → 34 passed).” This is from the mv p baseline

```
● (.venv) qamars-Laptop:task_manager_qamar qamarhareth$ pytest -q tests/test_integration.py
.....
11 passed in 0.07s
● (.venv) qamars-Laptop:task_manager_qamar qamarhareth$ pytest -q
.....
34 passed in 0.12s
● (.venv) qamars-Laptop:task_manager_qamar qamarhareth$ exit status
```

After I modified:

```
→ https://docs.pytest.org/en/stable/how-to/capture-warnings.html
60 passed, 2 warnings in 1.82s
❖ (.venv) qamars-Laptop:task_manager_qamar qamarhareth$ █
```

```
----- coverage: platform darwin, python 3.12.6-final-0 -----
Name          Stmts    Miss   Cover
-----
app.py          560     148    74%
config.py       21      0     100%
database.py     142     57     60%
gunicorn_config.py 23      23     0%
init_azure_sql.py 43      43     0%
tests/conftest.py 12      3     75%
tests/test_app.py 449     7     98%
tests/test_config.py 35      0     100%
tests/test_database.py 91      2     98%
tests/test_integration.py 126     7     94%
-----
TOTAL          1502    290    81%
60 passed, 2 warnings in 2.06s
❖ (.venv) qamars-Laptop:task_manager_qamar qamarhareth$ █
```

This is local

The screenshot shows the Taskly application interface on a local server. At the top, there are notifications: "You have 2 overdue tasks!" in red and "You have 1 task due today!" in green. The header includes a search bar, priority and category filters, and user information for "led" (led@gmail.com). Below the header, there are four main columns: "Overdue" (2 tasks), "Due Today" (1 task), "This Week" (1 task), and "Total Tasks" (3 total). The "Due Today" column contains a task titled "HPC assignment" with a due date of Dec 13, 11:18 PM. The "In progress" column contains a task titled "continue module 5" with a due date of Dec 13, 09:50 PM, marked as "Overdue".

The screenshot shows the Taskly application interface in dark mode. A modal window titled "Settings" is open, displaying options for "Theme" (App Theme: Dark Mode) and "Notifications" (Browser Notifications checked). The background shows the same task board structure as the first screenshot, with the "Due Today" column containing the "HPC assignment" task and the "In progress" column containing the "continue module 5" task.

Microsoft Azure Search resources, services, and docs (G+) Copilot

Home > qamar-taskmgr-db (task-manager-sql-8b/qamar-taskmgr-db) SQL database

Overview

Join us at **SQLCON Atlanta** from March 16-20 2026, for the ultimate SQL community learning event. Save \$200 with SQLAZP200.

Resource group (move) : BCSAI2025-DEVOPS-STUDENT-8B
Status : Online
Location : West Europe
Subscription (move) : Azure Simple I.E Instituto de Empresa, S.L.
Subscription ID : e0b9cada-61bc-4b5a-bd7a-52c606726b3b
Tags (edit) : Add tags

Server name : task-manager-sql-8b.database.windows.net
Connection strings : Show database connection strings
Pricing tier : General Purpose - Serverless: Gen5, 1 vCore
Auto-pause delay : 1 hour
Earliest restore point : 2025-12-14 00:34 UTC

Getting started Monitoring Properties Features Notifications (0) Integrations Tutorials

Start working with your database

Configure access Connect to application Start developing

Add or remove favorites by pressing Cmd+Shift+F

Microsoft Azure Search resources, services, and docs (G+) Copilot

Home > qamar-taskmgr-web Web App

Overview

Browse Stop Swap Restart Delete Refresh Download publish profile Reset publish profile Share to mobile

Resource group (move) : BCSAI2025-DEVOPS-STUDENT-8B
Status : Running
Location (move) : West Europe
Subscription (move) : Azure Simple I.E Instituto de Empresa, S.L.
Subscription ID : e0b9cada-61bc-4b5a-bd7a-52c606726b3b
Tags (edit) : Add tags

Properties Monitoring Logs Capabilities Notifications Recommendations

Web app

Name : qamar-taskmgr-web
Publishing model : Code
Runtime Stack : Python - 3.11
Runtime status : Healthy

Domains

Default domain : qamar-taskmgr-web-gyeraaeda3h5crg... Show More
Custom domain : Add custom domain

Microsoft Azure | portal.azure.com/#@teciehst.onmicrosoft.com/resource/subscriptions/e0b9cada-61bc-4b5a-bd7a-52c606726b3b/resourceGroups/BCSAI2025-D...

Home > qamar-taskmgr-web

qamar-taskmgr-web | Environment variables

Web App

App settings Connection strings

Name	Value	Deployment slot setting	Source	Delete
APPLICATIONINSIGHTS_CONNECTION_STRING	Show value		App Service	
AZURE_SQL_DATABASE	Show value		App Service	
AZURE_SQL_PASSWORD	Show value		App Service	
AZURE_SQL_SERVER	Show value		App Service	
AZURE_SQL_USERNAME	Show value		App Service	
DB_TYPE	Show value		App Service	
ENVIRONMENT	Show value		App Service	
FLASK_DEBUG	Show value		App Service	
SCM_DO_BUILD_DURING_DEPLOYMENT	Show value		App Service	

Add or remove favorites by pressing **Cmd+Shift+F**

Send us your feedback

Microsoft Azure | portal.azure.com/#@teciehst.onmicrosoft.com/dashboard/arm/subscriptions/e0b9cada-61bc-4b5a-bd7a-52c606726b3b/resourcegroups/bcsai20...

appi-qamar-taskmgr Dashboard

Shared dashboard

+ Create Upload Refresh Full screen | Edit Manage sharing Manage history Export Clone Assign tags Delete | Feedback

Last updated: a few seconds ago

Auto refresh: Off UTC Time: Past 24 hours

Get access to an improved dashboard experience and new mobile presence. [Try it now](#)

Usage: 5 Users

Reliability: Failed requests (Count), appi-qamar-taskmgr | 0

Responsiveness: Server response time

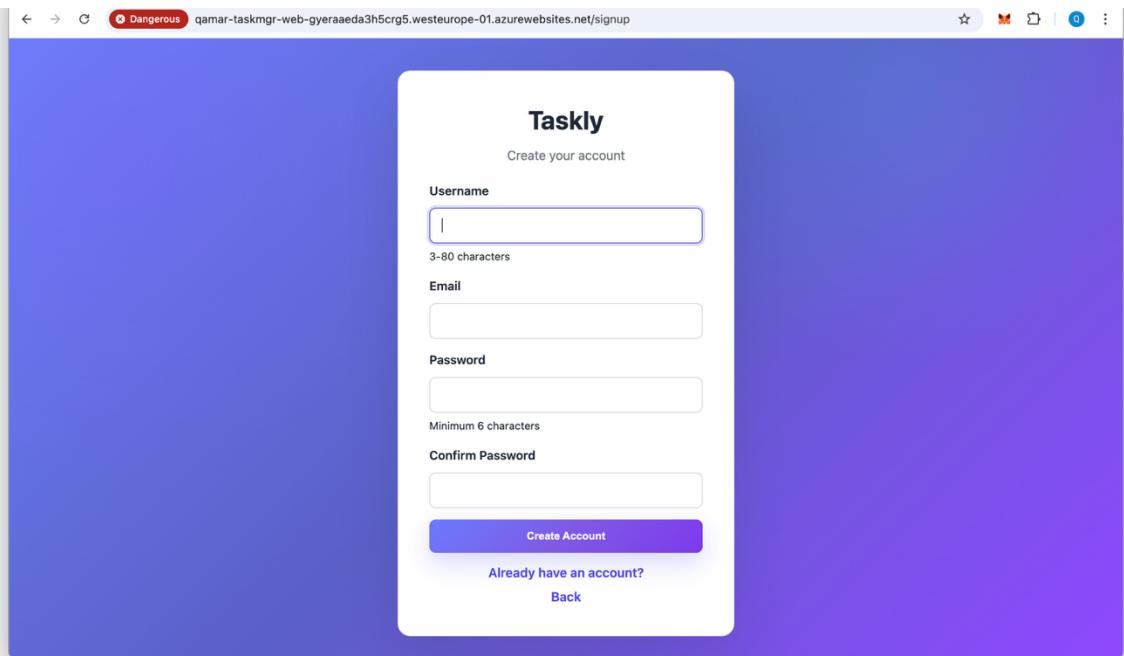
Browser: Average page load time breakdown

Average availability: 100%

Server exceptions and Dependency failures: 100%

Average processor and process CPU utilization: 100%

Browser exceptions: 100%



A screenshot of the Taskly task board interface. The top navigation bar shows the title 'Taskly' and a user profile for 'qamar'. Below the navigation is a search bar and filters for 'Priority' and 'Category'. The main area is a grid divided into four columns: 'Overdue', 'Due Today', 'This Week', and 'Total Tasks'. Each column contains a summary count (1, 1, 2, 3) and a list of tasks. A green callout bubble says 'Welcome back, qamar!'. Individual task cards show details like title, description, priority, category, due date, and status (e.g., 'To do', 'In progress', 'Done').

Taskly

Flow your tasks across the board

Search: Find by title or description

Priority: All priorities

Category: All categories

Overdue: 1

Due Today: 1

This Week: 2

Total Tasks: 3

To do: 1

In progress: 1

In review: 1

Done: 1

appointment for nie
go to the office
Medium Other
Due Dec 15, 09:15 AM
Due Today
Move to... Edit Delete

HPC Assignment
finish last 2 parts
High Education
Due Dec 16, 06:17 PM
Move to... Edit Delete

finish ai presentation
final touch up
Medium Education
Due Dec 14, 04:10 PM
⚠ Overdue
Move to... Edit Delete

Java
finish the slides
High Education
Due Dec 14, 06:22 PM
Move to... Edit Delete

Taskly

Flow your tasks across the board

Search: Find by title or description

Priority: All priorities

Category: Education

Overdue: 1

Due Today: 1

This Week: 2

Total Tasks: 3

To do: 0

In progress: 1

In review: 1

Done: 1

HPC Assignment
finish last 2 parts
High Education
Due Dec 16, 06:17 PM
Move to... Edit Delete

finish ai presentation
final touch up
Medium Education
Due Dec 14, 04:10 PM
⚠ Overdue
Move to... Edit Delete

Java
finish the slides
High Education
Due Dec 14, 06:22 PM
Move to... Edit Delete

Taskly

Flow your tasks across the board

Search: Find by title or description

Priority: High

Category: All categories

Apply, Reset, + New Task

Overdue: 1

Due Today: 1

This Week: 2

Total Tasks: 3

To do: 0

In progress: 1

In review: 0

Done: 1

HPC Assignment
finish last 2 parts
High Education
Due Dec 16, 06:17 PM
Move to... Edit Delete

Java
finish the slides
High Education
Due Dec 14, 06:22 PM
Move to... Edit Delete

Taskly

Flow your tasks across the board

Search: Find by title or description

Priority: All priorities

Notifications

Overdue: 1 task

Due Today: 1 task

Find by title or description

Priority: All priorities

Apply, Reset, + New Task

Overdue: 1

Due Today: 1

This Week: 2

Total Tasks: 3

To do: 1

In progress: 1

In review: 1

Done: 1

appointment for nie
go to the office
Medium Other
Due Dec 15, 09:15 AM
Due Today
Move to... Edit Delete

HPC Assignment
finish last 2 parts
High Education
Due Dec 16, 06:17 PM
Move to... Edit Delete

finish ai presentation
final touch up
Medium Education
Due Dec 14, 04:10 PM
Overdue
Move to... Edit Delete

Java
finish the slides
High Education
Due Dec 14, 06:22 PM
Move to... Edit Delete

This screenshot shows the Taskly Kanban board interface in a light theme. At the top, there are three tabs: 'System inspection summary', 'qamar-taskmgr-web - Micros', and 'Taskly - Kanban Board'. The main area displays a Kanban board with four columns: 'Overdue' (1 task), 'Due Today' (1 task), 'This Week' (2 tasks), and 'Done' (1 task). Each card contains task details like title, description, priority, category, due date, and status indicators (e.g., 'Overdue'). A user profile menu is open on the right, showing options: 'Change Username', 'Change Password', 'Settings', and 'Logout'. The top navigation bar includes a search bar, priority and category filters ('All priorities', 'All categories'), and a user icon.

This screenshot shows the Taskly Kanban board interface in a dark theme. A settings overlay is displayed in the center, titled 'Settings'. It contains two sections: 'Theme' (with 'App Theme' set to 'Dark Mode') and 'Notifications' (with a checked checkbox for 'Browser Notifications'). The background Kanban board is visible, showing the same four columns: 'Overdue' (1 task), 'Due Today' (1 task), 'This Week' (2 tasks), and 'Done' (1 task). The overall aesthetic is darker compared to the light theme version.

Taskly

Flow your tasks across the board

Search

Priority

Category

Apply Reset + New Task

Overdue 1

Due Today 1

This Week 2

Total Tasks 3

To do 1

appointment for nie

go to the office

Medium Other

Due Dec 15, 09:15 AM

Due Today

Move to... Edit Delete

In progress 1

HPC Assignment

finish last 2 parts

High Education

Due Dec 16, 06:17 PM

Move to... Edit Delete

In review 1

finish ai presentation

final touch up

Medium Education

Due Dec 14, 04:10 PM

Overdue

Move to... Edit Delete

Done 1

Java

finish the slides

High Education

Due Dec 14, 06:22 PM

Move to... Edit Delete

Find by title or description

Goodbye, qamar!

Taskly

Organize your tasks efficiently

Get Started Log in

Taskly © 2025

qamar-taskmgr-web-gyeraeda3h5crg5.westeurope-01.azurewebsites.net