Release Summary

Based on the following template/requirements:
https://drive.google.com/file/d/1hFMcawuigIK1hgGYVurLaupGsUORvEo6/view

Heading

Release Summary
Product name: WebWatch
Date: 12/03/24

Key user stories and acceptance criteria

- Users can create an account with an email and password
- Users can be notified through Email, Discord, and Slack whenever a web page is changed
- Web scraper should work for most web pages, without requiring additional programming to scrape certain sites
- Create an intuitive web application with a user-friendly GUI for managing tasks and notifications
- Users should be able to edit, pause/resume, and delete running tasks

Known Problems

- Lack of protections against email spam abuse:
  - Our free SMTP plan only allows for a few hundred emails per day, which is not viable for our product, as one frequently-notifying task could exhaust our email usage entirely. This would force us to choose between decreasing task run frequency or paying for more emails until we could implement more advanced rate limiting on a per-user basis.
  - We have disabled our email notification channel because of this
- Lack of email ownership verification
  - Currently, anyone can register an account using any email to abuse our service to spam someone's inbox. Currently, email sending is only for demonstration purposes and will be commented out from production code outside of our demo timeframes.
- The timezone used in scraping does not match the user's locale, causing scraped times to be different from the user's local time.
  - Possibly due to improper handling between UTC and local time.

- Our website does not scale well to different resolutions, making it well-suited for desktops but difficult to use on mobile devices.
- The task edit popup menu may be displaced when enough tasks are added to scroll.
- Webhooks in frontend do not have a retry mechanism on failure. If the server restarts/refreshes then the webhook will be disconnected and the user would have to refresh for their data to be streamed real-time
- Account Settings page does not have functionality to set global variables or link to OAuth providers
- Error messages for registration and account related components are not descriptive enough for the user
- Certain errors encountered during task scraping are added to the task's "content" field in the database, which causes it to be shown to the user the next time the task runs successfully and updates that content.
- Lack of detailed documentation for usage procedures such as finding a Discord/Slack webhook or XPath.
- Frontend reset password complexity requirements do not enforce the same requirements as the backend.
- Reset password currently only asks the user for their new password and to confirm their new password, but it would be better security practice to require the user to enter their current password before changing account information such as email or password.
- Some browsers show unnecessary scroll bars on our edit task component.
- Some text doesn't get parsed by BeautifulSoup.
- Some websites cause Selenium to hang and consume resources (msnbc.com)
- The deployment server is easily overwhelmed and causes scheduling disruptions.

Product Backlog

As a user, I would like a modular CLI service to interact with the backend.

- Moved from Sprint 1. We want to implement this as an additional feature in the future, but it's not important to our MVP at the moment.

I would like to receive notifications through multiple channels (email/discord/slack/etc) for detections.

- Moved from Sprint 1. We started this functionality in sprint 1, but it wasn't yet available as a completed feature to the end user.

As a user, I would like to be able to log in with external identity providers, like signing in with Google (OpenID/OAuth).

- Moved from Sprint 2. Implementing basic email/password authentication took longer than expected, so we didn't have time for this additional option.

As a user, I would like to be able to track the prices on online marketplaces so I can buy things cheaply.

- Moved from Sprint 3 for the same reasons as the previous story.

As a user, I would like to have the option to wire together my own custom action blocks in order to have the scheduler perform specific tasks I define.

- Moved from Sprint 4; we need all available time to finish implementing previous user stories, and this was not necessary for the minimum viable product

As a user, I would like to have the ability to create and share templates of my custom action workflow. That way, other users can extend my workflow and further customize it.

- Move from Sprint 4 for the same reasons as the previous story.