



WebWatch

Tracking Changes
Keeping You Updated

Cameron Candau (PO), Alex Kekchidis, Isaac To,
Jordan Nguy, Leonardo Gallego, Simon Zhao

December 4th, 2024

Motivation

Set out to Achieve:

Time-constrained individuals need timely alerts for specific events

For example, buying a parking permit from TAPS or waiting for course offerings to become available



Opportunity:

Automate process to provide the latest information while minimizing manual effort

Issue:

Manually checking for updates is time-consuming

Goals

Monitoring and Notification System

Users can be notified through Email, Discord, and Slack whenever a web page is changed

Robust User Account Features

Users can create accounts securely with an email and password, and reset their passwords when needed

Site-Agnostic Web Scraping

Web scraper should work for most web pages, without requiring additional programming to scrape certain sites

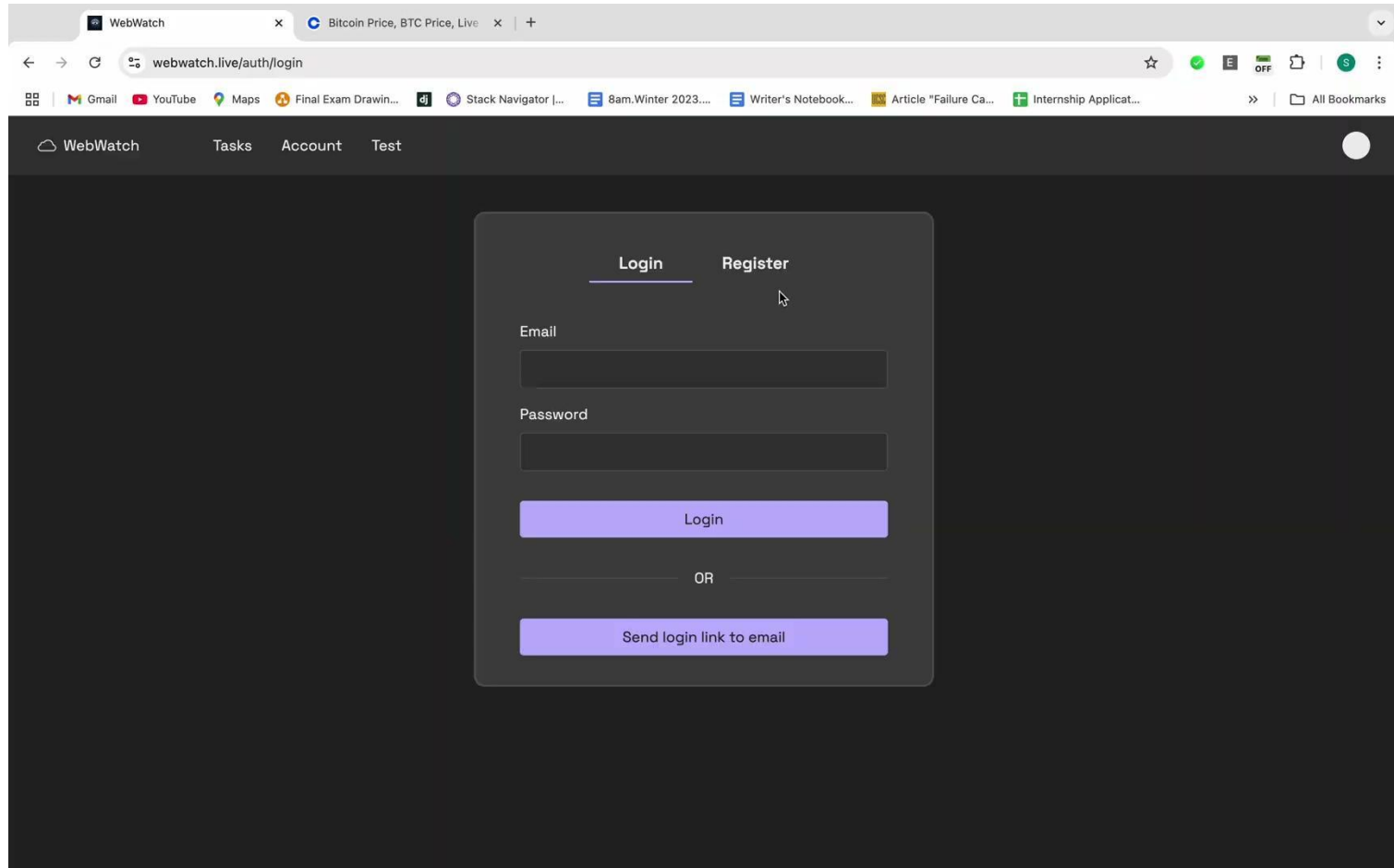
User-friendly Interface

Create an intuitive web application with a user-friendly and responsive GUI for managing jobs

Biggest Challenges

- **Insufficient Planning and Code Dependencies:**
 - Reliance on specific parts of the code without a defined interface, limiting asynchronous development
- **Team Communication and Coordination:**
 - Keeping track of team members' tasks, progress, and estimated timelines for better coordination
- **Accommodating More Common, Complex Use Cases:**
 - Many commercial websites use anti-botting mechanisms which block our current product and limit its utility.
 - Making the scraper generic enough to work for most sites, while still providing utility – just scrape visible text.

Demo



The screenshot shows a web browser window with the URL `webwatch.live/auth/login`. The browser's address bar and tabs are visible at the top. The page has a dark theme with a navigation bar at the top containing links for `WebWatch`, `Tasks`, `Account`, and `Test`. A central modal form is displayed with two tabs: `Login` (selected) and `Register`. The `Login` tab contains two input fields labeled `Email` and `Password`. Below these fields are two buttons: `Login` and `Send login link to email`. An `OR` separator is positioned between the `Login` button and the `Send login link to email` button. The `Register` tab is currently inactive.

WebWatch Tasks Account Test

Login Register

Email

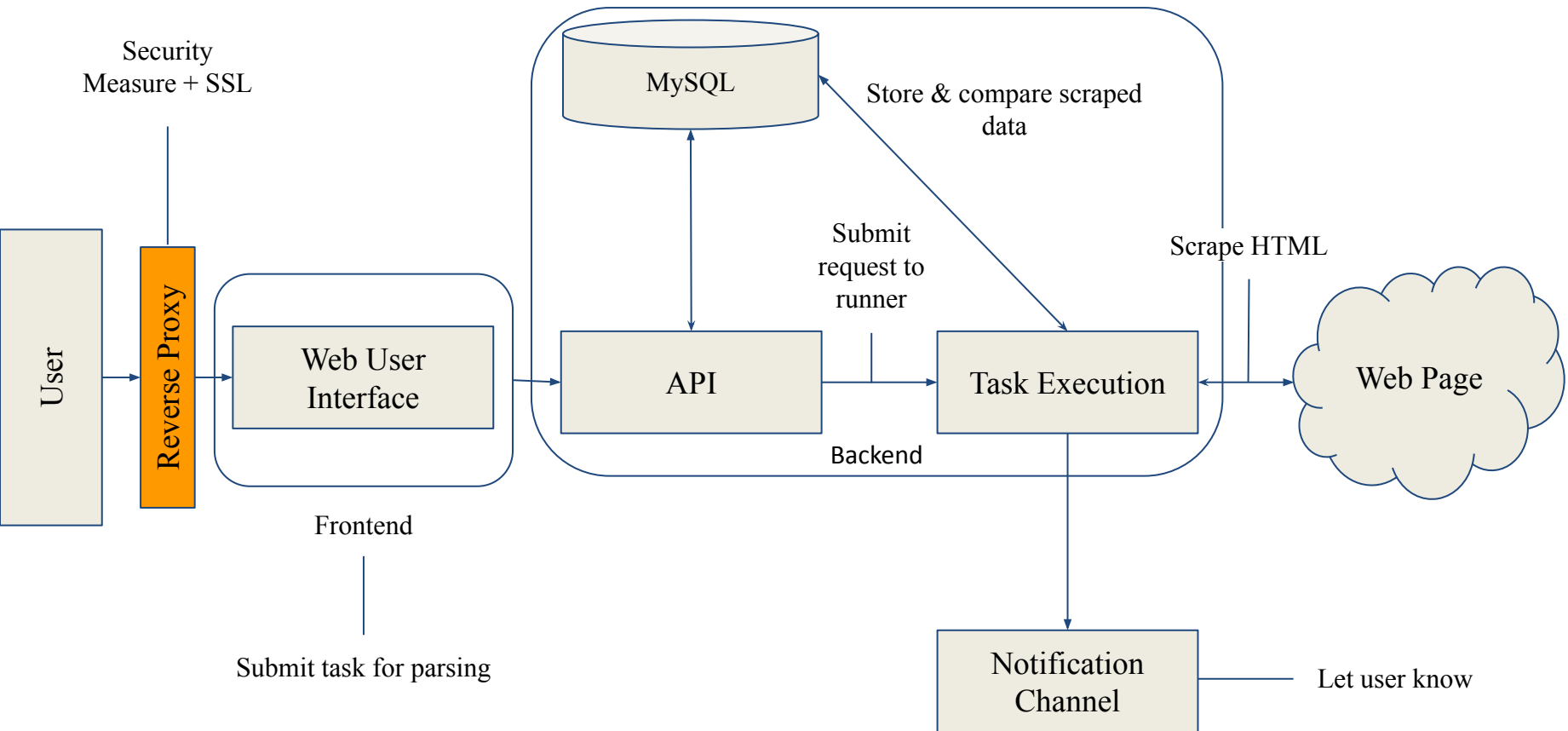
Password

Login

OR

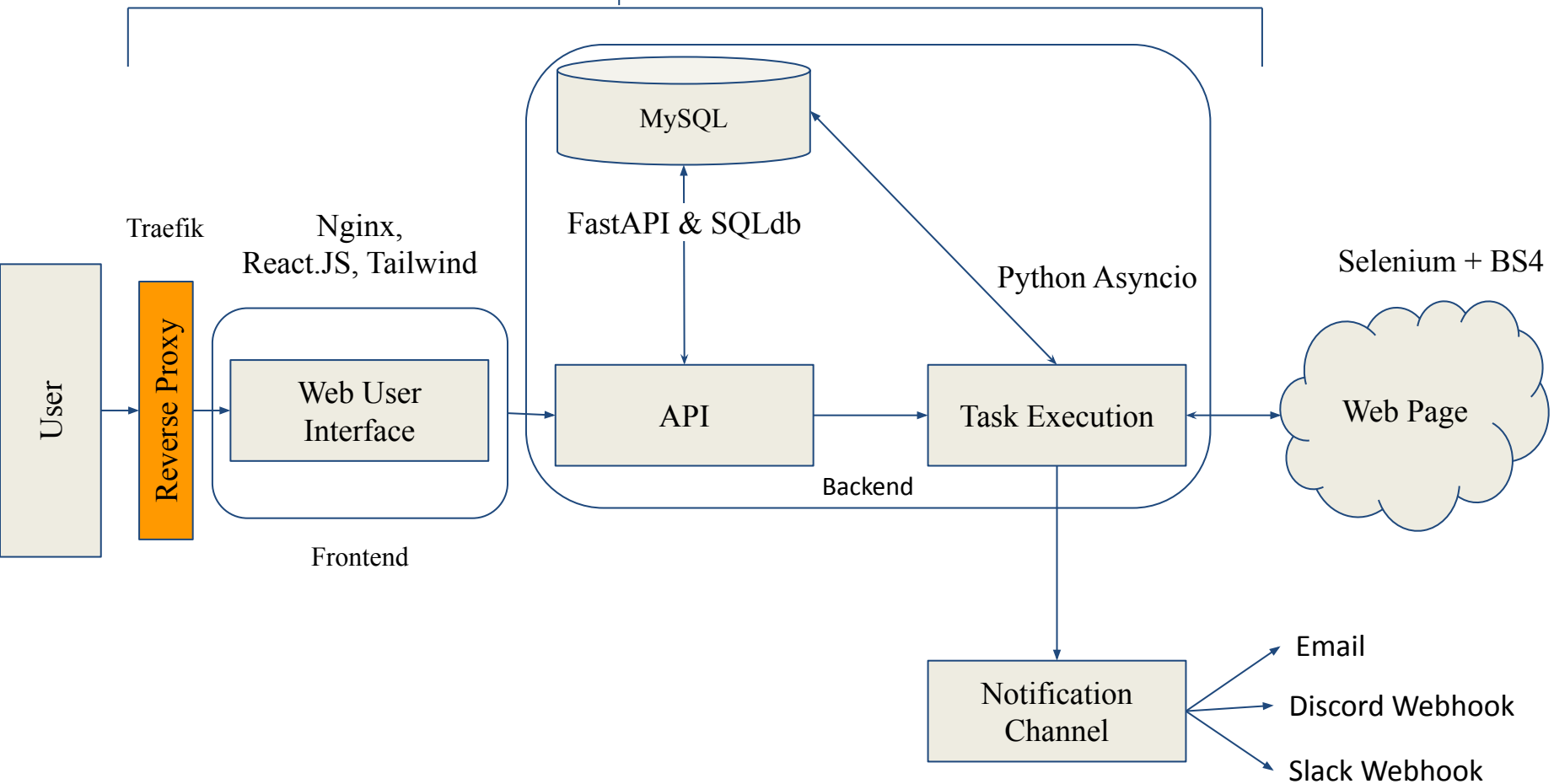
Send login link to email

System overview/diagram



Technologies

Docker for Containerization & Deployment



Project Management Technique

Scrum Practices

- **Sprints:** Structured planning and review cycles
- **Jira:** Scrum board and burnup charts for progress tracking
- **Meetings:** Standard weekly times for team sync
- **Leadership:** Led by a dedicated Scrum Master
- **Documentation:** Detailed meeting notes
- **Version Control:** Feature branches with code reviews

Deviations from Scrum:

- Minimal planning upfront
- Frequent task creation throughout sprints



Reflections on the Project

- **What We Enjoyed:**

- Working with each other as a team
- Experiencing Scrum framework for the first time
- Learning new technologies/frameworks

- **What We Didn't Enjoy:**

- Debugging asynchronous errors within task scheduling
- Prioritizing user stories and functionality over code quality; resulted in increased technical debt

Valuable Lessons

- **What worked?**
 - Using a scrum board to coordinate work
 - Containerization to avoid version conflicts
- **What didn't work?**
 - Lack of structured code review process, causing delay in integration
- **What We Wish We Had Done:**
 - More generous, larger estimates when creating user stories and tasks
 - Better communication beyond weekly meetings