Google Summer of Code

2025

Personal Information:

- **Name:** Piumal Rangana
- **GitHub:** https://github.com/PiumalRangana
- **Email:** piumalrangana4@gmail.com
- **Resume:** Resume new.docx
- **Time Commitment:** Available for at least 5 hours daily, with flexibility for extra hours

---

Project:

📅 Releases Working Group Calendar Website

Organization:

 Electron

Expected Outcomes:

This project aims to automate the tracking of Chromium's release schedule and build a web-based release calendar for the Electron team. The main goals are:

- **Automate** fetching Chromium release dates using their API.
- **Develop** an intuitive and user-friendly web interface.
- **Enhance** automation by integrating bots or webhooks to notify the team when dates change.

This solution will improve Electron's release management and reduce the manual work required for tracking updates.

Benefits to the Community:

The Electron community depends on Chromium's release schedule, but currently, updates are tracked manually. Automating this process will:

- Ensure **real-time updates** without manual intervention.
- Reduce human errors in tracking Chromium versions.
- Provide an **interactive release calendar** for better visualization.
- Improve **team communication** with automated Slack alerts

By streamlining this process, the Electron team can focus more on development instead of manual tracking.

Deliverables

Phase 1: Data Automation

- Write scripts to fetch Chromium release dates using their API.
- Store and structure fetched data in an efficient format using a **relational database**.

Phase 2: Release Calendar Website

- Develop a **barebones frontend** to display release dates.
- Implement an **Express.js backend** to serve the release dates.
- Design the website using **HTML, CSS, and JavaScript**.

Phase 3: Automated Update Mechanism

- Implement a scheduled process to fetch Chromium release data every **six hours**.
- Compare the new data with stored data in the **DataBase**.
- If changes are detected, update the calendar and send **Slack notifications** via a webhook to `#wg-releases`.
- Provide an interface for Electron team members to manually trigger an update check.

Phase 4: Deployment & Documentation

- Deploy the release calendar on `releases.electronjs.org`.
- Write detailed documentation for future contributors.

Technical Details

Tech Stack

Back end	Express.js, Node.js
Frontend	HTML, CSS, JavaScript (or React if preferred)
Database	Relational Database(SQlite or MySql,PostgreSQL)
Automation	Scheduled updates every six hours, Slack API notifications, manual update button
External API	Chromium API

Fetching Chromium Release Data

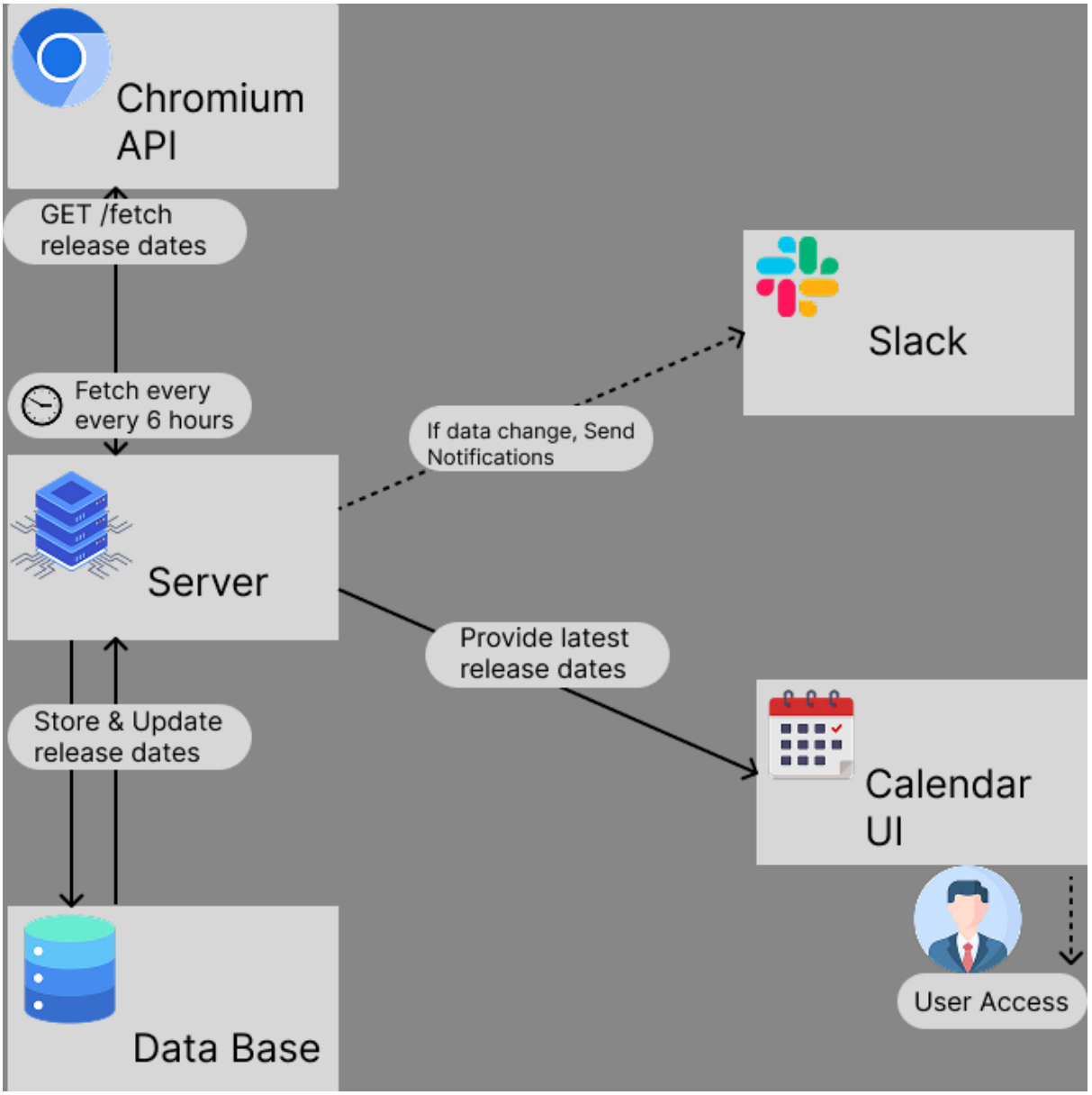
- **API URL** allows fetching as much data as needed in one request.

- The fetched data will be stored in a **Relational Database** for structured storage and querying.

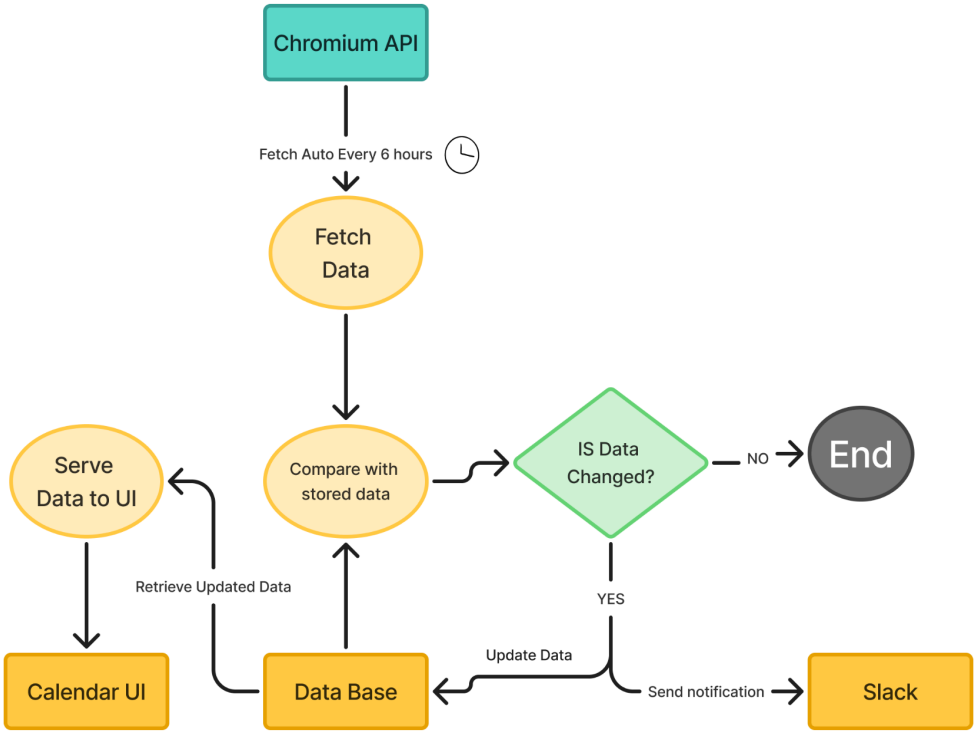
Workflow Automation

- **Scheduled Fetching**: Every **six hours**, fetch Chromium release data and check for updates.
- **Manual Check Interface**: Allows Electron team members to fetch new release dates on demand.
- If changes are detected, **update the calendar** and send a **Slack notification**.

Below is the system architecture illustrating the automation process:



The following Data Flow Diagram illustrates how data is fetched, stored, compared, and updated in the system.



Timeline

Period	Task
Community Bonding	Understand Electron’s release process, engage with mentors.
Week 1-2	Set up the project structure, research Chromium API.
Week 3-4	Implement Chromium API data fetching with error handling and caching.
Week 5-6	Develop backend routes for the calendar.

Week 7-8	Build the frontend UI for displaying release dates.
Week 9-10	Implement automated update mechanism and Slack notifications.
Week 11-12	Finalize, test, document, and deploy the project.

Expected Challenges

1. Chromium API Limitations

- API rate limits or changes → **Solution:** Ensure robust error handling and caching.

2. Keeping the Calendar Updated

- Chromium release dates might change → **Solution:** Implement scheduled updates every six hours and a manual update button.

3. Slack Integration

- Configuring Slack webhooks might require permissions → **Solution:** Work with mentors to get API access.

Future Work

- **Enhancing the Automation Process**
  - Improve the scheduling system to optimize API calls and reduce redundant checks.
  - Explore alternative ways to detect release changes with lower latency.

- **Expanding Slack Bot Features**
  - Allow team members to query upcoming releases directly from Slack.
  - Implement custom notification settings (e.g., alerts for specific release types).

- **Maintaining and Extending the Project**
    - Actively participate in issue discussions and PR reviews for further improvements.
    - Continue to refine the UI and backend based on user feedback.

Resources

- Chromium API Docs
- Electron Releases Working Group
- Slack API Docs

Why Me?

- **Experience in Web Development:** Skilled in JavaScript, TypeScript, Python, Angular and Express.js.

- **Automation Enthusiast:** Previously built web scrapers and interested in bots/webhooks.

- **Strong Commitment:** Can dedicate extra hours to ensure project success.

- **GSoC Goal:** Contribute meaningfully to open source while improving my skills.

Open to feedback and improvements before submission!

Thank you…