Google Calendar Sync Project

AUTHOR
Jayden Carter, Nathan Ho, Tommy Henry, Yahir Moreno

PUBLISHED
March 12, 2025

Overview:

Students often struggle with managing assignments across multiple learning platforms like Canvas and Gradescope. The lack of a centralized system leads to missed deadlines and inefficiencies due to manual tracking. In this project we automated assignment tracking using a Chrome extension that extracts assignment data and then syncs it to Google Calendar through a Python Django backend API we developed to call the Google Calendar API (This was done for the sake of our course being in python not for efficiency).

Objective:

The **Student Calendar Syncer** is a Google Chrome extension that is designed to automate assignment tracking. Extracting assignment data from Canvas and Gradescope, it stores them locally, and then syncs them with Google Calendar through a Django back-end API to automatically schedule assignments. This automation allows for students to have an up-to-date centralized system that will help manage their academic workload.

Project Scope

There are three main components:

1. **Chrome Extension** - Extracts assignments, sends data to backend, and allows for user interaction.
2. **Django Backend** - Processes extracted assignments, stores data locally to chrome local storage, validates the data, and then syncs it with Google Calendar.
3. **Google Calendar Integration** - Assignments are scheduled automatically as events.

Methodology:

System Overview: The system follows a structure that processes assignments dynamically.

1. **Assignment Extraction:** content.js scrapes assignments from Canvas and Gradescope. This script is injected into Canvas and Gradescope when a user visits their assignment page. It extracts assignment details, formats them into JSON, and then sends the data to background.js for storage and processing.
2. **Temporary Storage:** Assignments are stored in chrome.storage.local.
3. **Data Transmission:** background.js sends extracted assignments to Django. The background script (background.js) stores, validates, and syncs assignment data to Django (/api/assignments/sync/), logs actions, and ensures smooth extension-backend communication.
4. **Processing/Validation:** Django API validates and processes assignments. This serves as the middle layer between Chrome and Google Calendar.

Google Calendar Sync: Assignments are converted into calendar events via ProcessAssignments.py

```python
def getOrCreateCalendar(service, className):
    """
    Checks if a calendar for the class exists, otherwise creates a new one.
    This function Returns the calendar ID :).
    """
    try:
        # Fetch existing calendars
        calendarList = service.calendarList().list().execute()

        # Check if the calendar already exists
        for calendar in calendarList.get("items", []):
            if calendar["summary"] == className:
                print(f"📖 Found existing calendar: {className}")
                return calendar["id"]  # Return existing calendar ID

        # If not found, create a new calendar
        new_calendar = {
            "summary": className,
            "timeZone": "America/Los_Angeles"
        }
        created_calendar = service.calendars().insert(body=new_calendar).execute()
        print(f"📅 Created new calendar: {className}")

        return created_calendar["id"]

    except HttpError as error:
        print(f"❌ Error checking/creating calendar for {className}: {error}")
        return "primary"  # Default to primary if error occurs
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
venvjaydencarter@169-231-105-125 PSTAT134FinalProject % cd studentCalander
venvjaydencarter@169-231-105-125 studentCalander % python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
March 13, 2025 - 22:10:14
Django version 5.1.7, using settings 'studentCalander.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.

)
```

- **Chrome Extension** :

    1. **Manifest.JSON:** Configuration and Permissions

        - defines permissions and specfies which sites the extension can interact with

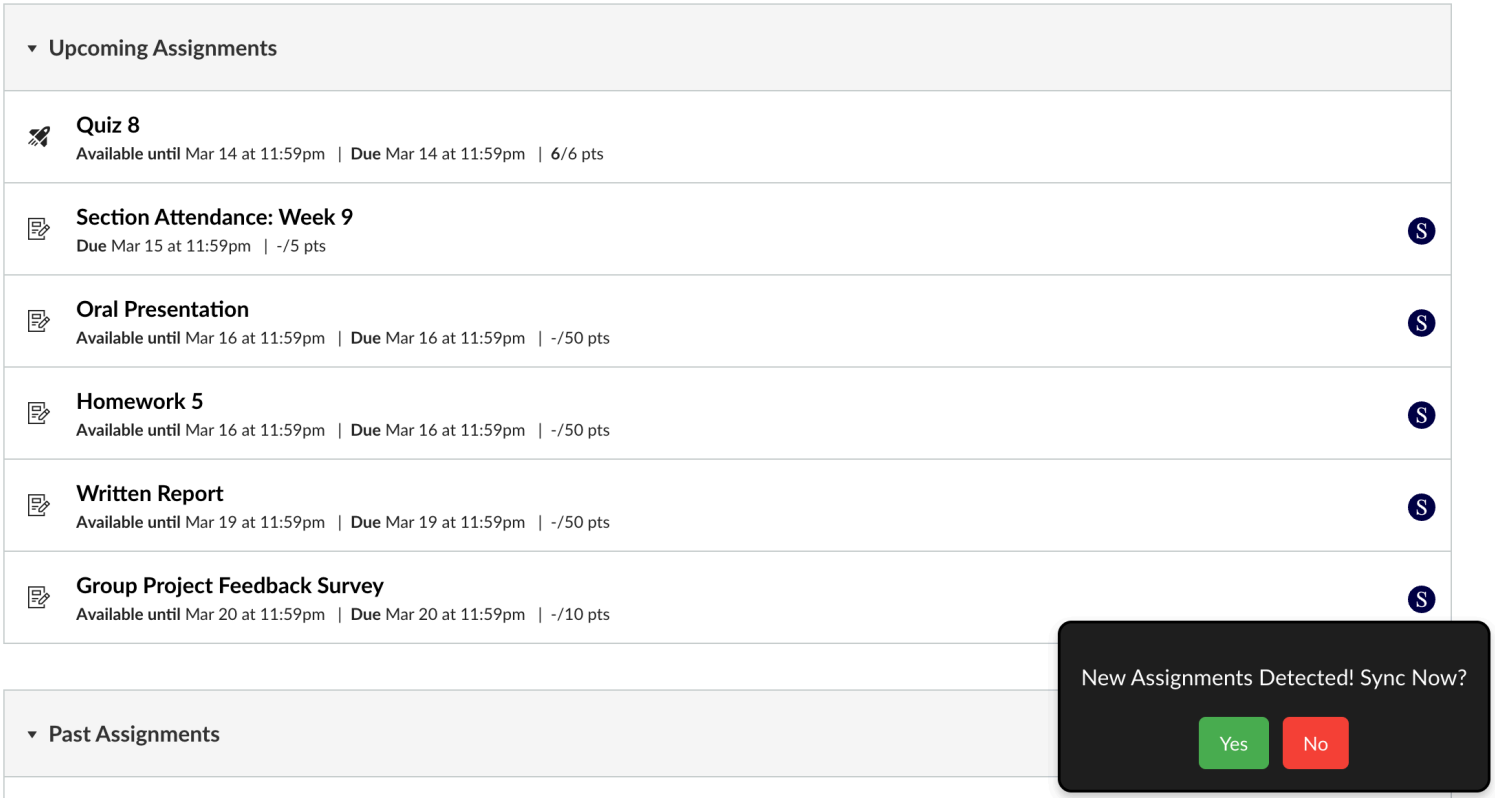
    2. **Content.js:** Assignment Extraction

        - injected into Canvas and Gradescope when the assignment page loads. Extracting assignment details like Title, Due Date, Points, and Assignment Link.

        - stores extracted data in chrome.storage.local before sending it to background.js for further processing

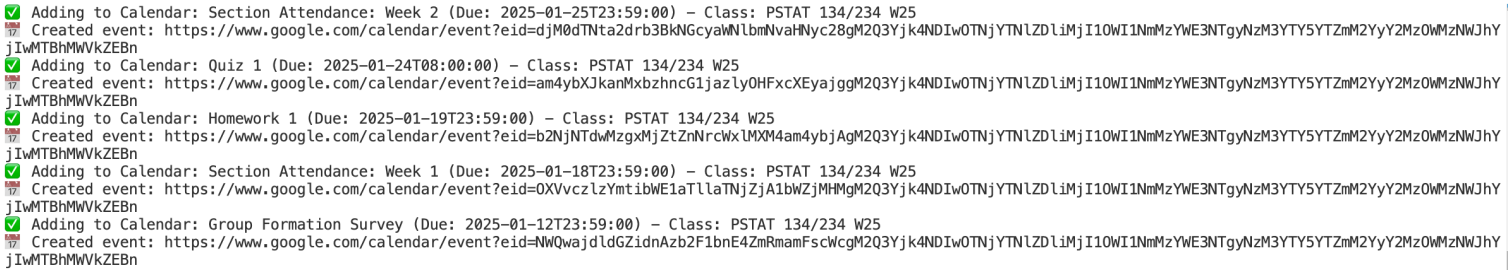
- **Background.js -** Data Processing and API Communication - Retrieves stored assignments from chrome.storage.locally - This validates and formats assignment data. - Sending a POST request to the Django API (/api/events) to sync assignments. - Logging all assignment sync transactions showing a successful data transmission to Django.

- **Popup.js -** Unused in this project



- **Django Backend**
- The Django backend acts as the intermediary between the Chrome Extension and the Google Calendar, ensuring that assignment data is processed correctly before being synced.
    1. **Views.py – API Request Handling**
        - Receives assignment data from background.js via a POST request.
        - Validates the data to ensure correct formatting.
        - Stores assignments in the Django database.
        - Provides a GET request to retrieve stored assignments if needed. Insert Screenshot: API response log confirming successful assignment storage.

2. **ProcessAssignments.py – Google Calendar Integration**
    - Authenticates via OAuth2 to securely access the user's Google Calendar.
    - Creates Google Calendar events based on the received assignment details.
    - Prevents duplicate assignments by checking for existing calendar events.



3. **Serializers.py – Data Serialization and Validation**
    - Converts Django model instances into JSON format and vice versa.
    - Handles automatic field management using serializers.ModelSerializer.
    - Ensures data is structured and validated before being processed.
    - Prevents incomplete or incorrectly formatted data from entering the system.

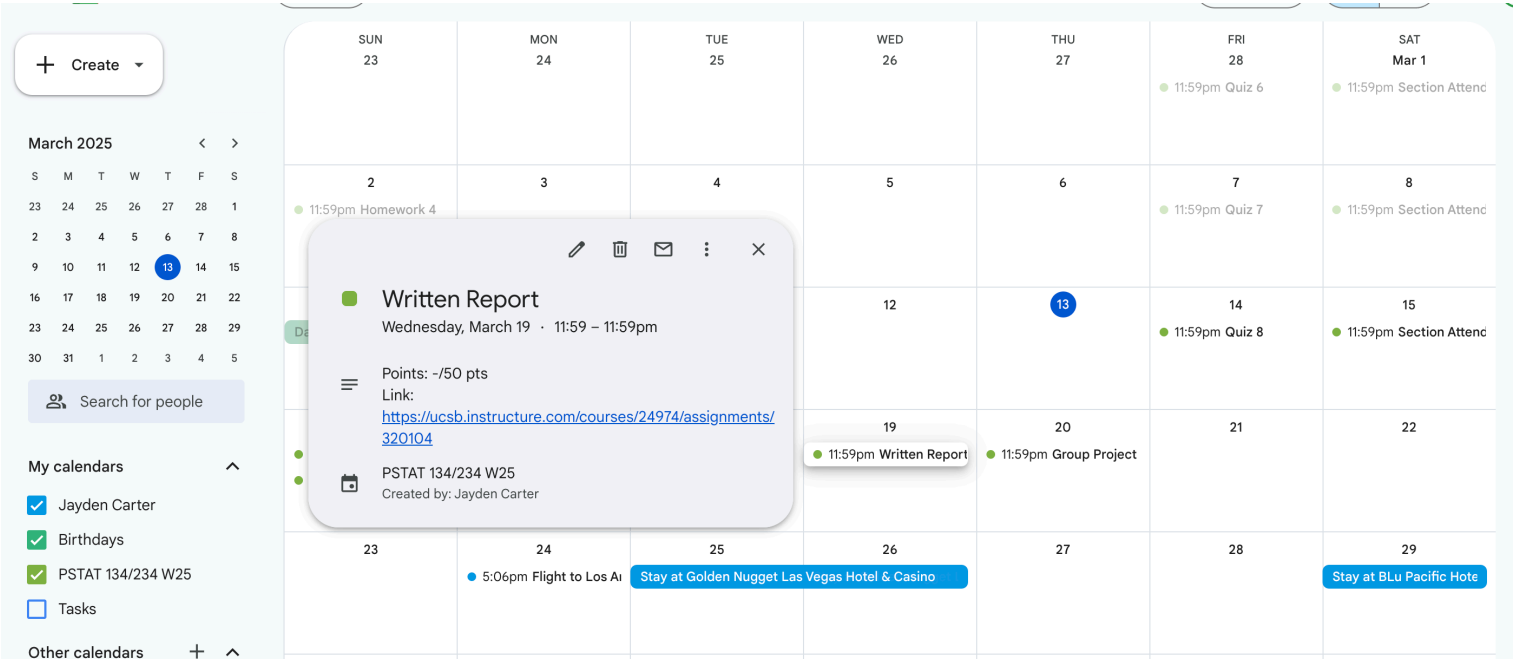
Results and Testing: Testing was conducted at multiple stages to ensure accuracy and reliability.

This was tested on all classes and assignments we had available to us on both our canvas and gradescope for our current and past classes.

1. Assignment Extraction - Assignments were successfully scraped from Canvas and Gradescope. - Data was stored correctly in chrome.storage.local Insert Screenshot: Console log of extracted JSON data from assignments.

2. API Data transmission - Django API properly received and processed assignment data.

3. Google Calendar Syncing - Assignments correctly appeared in Google Calendar with the right title and due date.



Conclusion: In this project we successfully automated the process of extracting and syncing assignments across platforms, eliminating the need for manual tracking. This system is fully functional but can be improved. Improvements include removing Python for full JavaScript automation, data duplication handling, syncing gradescope properly with links, and including other calendar websites like Notion and Microsoft Outlook as well.

LEBRON JAMES
2022 WALL CALENDAR
BONUS! Includes September - December 2021
LEBRON JAMES
LOS ANGELES LAKERS
SEP 2021
OCT 2021
NOV 2021
DEC 2021
Cover photo by Adam Pantozzi/NBAE via Getty Images
The publisher has made every effort to ensure the accuracy of the data presented in this calendar edition. We cannot be held responsible for any errors, omissions or inconsistencies.
Please Note: All featured players are under contract at time of printing.

Made in China
U.S. $9.99
50999

TURNER
by design group

NBA.com
Item #22998040625