**Study Schedule**

## Abstract

Creating a website that displays interactive graphs for machine learning results and connects to a database or SQL server involves several layers of development, including the backend, database, front end, and machine learning components.

To develop a web-based dashboard that performs real-time sentiment analysis on social media posts related to a specific topic or hashtag.

To accept Twitter API accessed information to determine the sentiment on the reception of the new videogame ‘Lies of P’ developed by Round 8 Studios, split into three categories; negative, neutral, and positive.

**Datasets:**

1. **Twitter API**: You can fetch real-time tweets based on hashtags or topics.
   * Source: [Twitter Developer Platform](https://developer.twitter.com/en/docs/twitter-api)
2. **Sentiment 140**: A dataset containing 1.6 million tweets labelled as positive, neutral, or negative.
   * Source: [Kaggle - Sentiment140](https://www.kaggle.com/kazanova/sentiment140)

# Phase 1: Planning & Setup

1. **Requirement Analysis:**
   * **Features:** Real-time data scraping, sentiment scoring, interactive graphs
   * **Machine Learning Models:** Text classification models for sentiment analysis
   * **User Interactions:** Users should be able to input a hashtag or topic and view sentiment over time.
2. **Environment Setup:**
   * Install Python, Flask, and Scikit-learn for the backend.
   * Set up MySQL database and install SQL Alchemy.
   * Initialize a Vue project with TypeScript and Plotly.js for the front end.

**Total Time Allocation:** 3-4 weeks

## Concepts:

### Real-Time Data Scraping

* + **What to Study:** How to scrape data from social media platforms in real time.
  + **Time Needed:** 3 days
  + **Resources:**
    - [Twitter API Documentation](https://developer.twitter.com/en/docs/twitter-api)
    - ["Web Scraping using Python" by Ryan Mitchell](https://www.amazon.com/Web-Scraping-Python-Comprehensive-Collecting/dp/1491985577/)
    - [Beautiful Soup Tutorials](https://realpython.com/beautiful-soup-web-scraper-python/)

### Sentiment Scoring Algorithms

* + **What to Study:** Basics of sentiment analysis algorithms and Python implementation.
  + **Time Needed:** 4 days
  + **Resources:**
    - ["Natural Language Processing" (Coursera)](https://www.coursera.org/specializations/natural-language-processing)
    - ["Applied Text Analysis with Python" (Book)](https://www.amazon.com/Applied-Text-Analysis-Python-Language/dp/1491963042/)
    - [Scikit-learn's text classification guides](https://scikit-learn.org/stable/supervised_learning.html#supervised-learning)
    - [(45) Python Sentiment Analysis Project with NLTK and 🤗 Transformers. Classify Amazon Reviews!! - YouTube](https://www.youtube.com/watch?v=QpzMWQvxXWk)

### Interactive Graphs for Dashboards

* + **What to Study:** Technologies for rendering interactive graphs on web dashboards.
  + **Time Needed:** 3 days
  + **Resources:**
    - [Plotly Documentation](https://plotly.com/python/)
    - [D3.js Tutorials](https://d3js.org/)

## Facts:

### Real-Time Analytics for Business Intelligence

* + **What to Study:** Importance of real-time analytics in business.
  + **Time Needed:** 2 days
  + **Resources:**
    - ["Real-Time Analytics: Techniques to Analyze and Visualize Streaming Data" (Book)](https://www.amazon.com/Real-Time-Analytics-Techniques-Visualize-Streaming/dp/1118824822)
    - [Articles on Towards Data Science](https://towardsdatascience.com/)
    - [Harvard Business Review articles](https://hbr.org/)

### Machine Learning in Sentiment Analysis

* + **What to Study:** Role of machine learning models in sentiment analysis.
  + **Time Needed:** 2 days
  + **Resources:**
    - [Research papers on arXiv](https://arxiv.org/)
    - ["Pattern Recognition and Machine Learning" (Book)](https://www.amazon.com/Pattern-Recognition-Learning-Information-Statistics/dp/0387310738)

## Procedures:

### How to Obtain Data from Twitter API

* + **What to Study:** Steps to register for a Twitter Developer Account and use Python libraries like Tweepy.
  + **Time Needed:** 1 day
  + **Resources:**
    - [Tweepy Documentation](https://docs.tweepy.org/en/latest/)
    - [Twitter Developer Platform Guides](https://developer.twitter.com/en/docs/tutorials)

### How to Preprocess and Store Streaming Data

* + **What to Study:** Data preprocessing techniques and real-time data storage methods.
  + **Time Needed:** 2 days
  + **Resources:**
    - ["Data Preprocessing for Machine learning in Python" (DataCamp)](https://www.datacamp.com/community/tutorials/preprocessing-in-data-science-part-1-centering-scaling-and-knn)
    - ["Streaming Systems" (Book)](https://www.amazon.com/Streaming-Systems-Where-Large-Scale-Processing/dp/1491983876/)

# Phase 2: Backend Development (Python with Flask)

1. **Initialize Flask Project:**
   * **Command**: Use Flask or a similar command to scaffold a new Flask project.
   * **Installation**: Install Flask-RESTful using **pip install Flask-RESTful**.
2. **Database Connection:**
   * **Configuration**: Use SQL Alchemy to connect with your MySQL database.
3. **API Endpoints:**
   * **Creation**: Use Flask-RESTful to create API endpoints.
   * **Example**: **/api/sentiment**
4. **Integrate Scikit-learn:**
   * **Integration**: Use text classification models from scikit-learn for sentiment analysis.
5. **Test Backend:**
   * **Testing**: Use Postman to test the API endpoints.

**Total Time Allocation**: 3-4 weeks

Concepts:

### Flask Framework

* **What to Study**: Understanding Flask and its capabilities.
* **Time Needed**: 1 week
* **Resources**:
  + [Flask Official Documentation](https://flask.palletsprojects.com/en/2.1.x/)

### RESTful APIs

* **What to Study**: Basics of RESTful APIs.
* **Time Needed**: 3 days
* **Resources**:
  + [RESTful API Quick Guide](https://www.tutorialspoint.com/restful/index.htm)

### MVC Architecture

* **What to Study**: Model-View-Controller architecture.
* **Time Needed**: 3 days
* **Resources**:
  + [MVC Architecture in 3 minutes](https://www.youtube.com/watch?v=DUg2SWWK18I)

## Facts:

### Benefits of RESTful APIs in Backend Development

* **What to Study**: Advantages of using RESTful APIs.
* **Time Needed**: 2 days
* **Resources**:
  + [Why RESTful APIs](https://restfulapi.net/)

### Usefulness of Postman in API Testing

* **What to Study**: Importance and advantages of using Postman for API testing.
* **Time Needed**: 2 days
* **Resources**:
  + [Getting Started with Postman](https://learning.postman.com/docs/getting-started/introduction/)

## Procedures:

### How to Initialize a Flask Project

* **What to Study**: Steps to create a new Flask project.
* **Time Needed**: 3 days
* **Resources**:
  + [Flask Quickstart Guide](https://flask.palletsprojects.com/en/2.1.x/quickstart/)

### How to Set Up Flask-RESTful

* **What to Study**: How to install and configure Flask-RESTful.
* **Time Needed**: 3 days
* **Resources**:
  + [Flask-RESTful Official Documentation](https://flask-restful.readthedocs.io/en/latest/)

### Steps to Test API Using Postman

* **What to Study**: How to test API endpoints with Postman.
* **Time Needed**: 2 days
* **Resources**:
  + [Postman Testing Guide](https://learning.postman.com/docs/writing-scripts/test-scripts/)

# Phase 3: Database Design & Setup (MySQL with SQLAlchemy)

1. **Design Database Schema:**
   * **Tables**: Create tables for storing tweets and sentiment scores.
2. **Data Insertion:**
   * **Insertion**: Use SQLAlchemy to insert data into MySQL.
3. **Data Retrieval:**
   * **Queries**: Implement queries to fetch sentiment data.

**Total Time Allocation**: 2-3 weeks

## Concepts:

### Relational Database Management System (RDBMS)

* **What to Study**: Fundamentals of RDBMS.
* **Time Needed**: 5 days
* **Resources**:
  + [RDBMS Fundamentals](https://www.guru99.com/rdbms-concepts.html)

### Database Schemas

* **What to Study**: What are database schemas and how to design them.
* **Time Needed**: 3 days
* **Resources**:
  + [Schema in Database](https://www.geeksforgeeks.org/sql-ddl-dml-dcl-tcl-commands/)

### SQL Queries

* **What to Study**: Basics of SQL queries for CRUD operations.
* **Time Needed**: 3 days
* **Resources**:
  + [SQL Tutorial](https://www.w3schools.com/sql/)

### Database Transactions

* **What to Study**: Understanding ACID properties in databases.
* **Time Needed**: 2 days
* **Resources**:
  + [ACID Transactions](https://en.wikipedia.org/wiki/ACID)

## Facts:

### Importance of Efficient Database Design

* **What to Study**: Why efficient database design matters.
* **Time Needed**: 2 days
* **Resources**:
  + [Database Design Article](https://www.essentialsql.com/what-is-a-database-schema/)

### How Database Indexes Work

* **What to Study**: The role and importance of indexes in databases.
* **Time Needed**: 2 days
* **Resources**:
  + [Database Indexing](https://www.essentialsql.com/what-is-the-difference-between-a-heap-and-clustered-table/)

## Procedures:

### How to Design a Database Schema

* **What to Study**: Steps involved in designing a database schema.
* **Time Needed**: 4 days
* **Resources**:
  + [Database Design Tutorial](https://www.datanamic.com/support/lt-dez005-introduction-db-modeling.html)

### How to Use SQLAlchemy for CRUD Operations

* **What to Study**: How to perform CRUD operations using SQLAlchemy.
* **Time Needed:** 4 days.
* **Resources**: [SQLAlchemy Tutorial](https://auth0.com/blog/sqlalchemy-orm-tutorial-for-python-developers/)

### Setting up triggers and stored procedures in MySQL

* **Time Needed:** 2 days.
* **Resources**: [MySQL Triggers](https://dev.mysql.com/doc/refman/8.0/en/triggers.html)

# Phase 4: Frontend Development (Vue with Plotly.js and TypeScript)

1. **Initialize Vue Project:**

* **Project Setup**: Use Vue CLI to initialize a new Vue project with TypeScript and Plotly.js.

1. **UI/UX Design:**

* **Wireframes**: Sketch out wireframes to determine how the dashboard should look and feel.

1. **Interactive Graphs:**

* **Visualization**: Utilize Plotly.js to visualize sentiment data as interactive graphs.

1. **API Integration:**

* **Data Fetching**: Use Axios to interact with Flask API endpoints and retrieve data.

1. **User Testing:**

* **Quality Assurance**: Make sure the front end is responsive and free of errors.

**Total Time Allocation**: 4-6 weeks

## Concepts:

### Vue.js Framework

* **What to Study**: Understand the basics and capabilities of Vue.js.
* **Time Needed**: 1 week
* **Resources**:
  + [Vue.js Guide](https://vuejs.org/v2/guide/)

### Plotly.js for Data Visualization

* **What to Study**: Learn how to use Plotly.js for creating interactive graphs.
* **Time Needed**: 1 week
* **Resources**:
  + [Plotly.js Documentation](https://plotly.com/javascript/)

### TypeScript

* **What to Study**: Grasp the basics of TypeScript for improved code quality.
* **Time Needed**: 1 week
* **Resources**:
  + [TypeScript Handbook](https://www.typescriptlang.org/docs/handbook/intro.html)

## Facts:

### Advantages of Interactive Graphs for Data Visualization

* **What to Study**: The benefits and use-cases of interactive graphs.
* **Time Needed**: 2 days
* **Resources**:
  + [Interactive Data Viz Article](https://towardsdatascience.com/interactive-controls-for-jupyter-notebooks-f5c94829aee6)

### Importance of Responsive Design

* **What to Study**: The significance of making a web application responsive.
* **Time Needed**: 2 days
* **Resources**:
  + [Responsive Web Design](https://www.w3schools.com/css/css_rwd_intro.asp)

### Web Accessibility Guidelines

* **What to Study**: Understanding web accessibility standards.
* **Time Needed**: 1 day
* **Resources**:
  + [W3C Guidelines](https://www.w3.org/WAI/standards-guidelines/wcag/)

## Procedures:

### How to Set Up a Vue Project with TypeScript and Plotly.js

* **What to Study**: The step-by-step procedure to initialize a Vue project.
* **Time Needed**: 5 days
* **Resources**:
  + [Vue.js with TypeScript Setup](https://vuejs.org/v2/guide/typescript.html)

### Steps to Integrate APIs in Frontend Using Axios

* **What to Study**: How to call and integrate APIs using Axios in a Vue.js application.
* **Time Needed**: 5 days
* **Resources**:
  + [Axios with Vue](https://vuejs.org/v2/cookbook/using-axios-to-consume-apis.html)

# Phase 5: Integration & Testing

Tasks:

1. **Full Stack Integration:**
   * **Communication**: Ensure the frontend and backend can communicate correctly.
2. **Functional Testing:**
   * **Data Flow**: Test the entire flow of data from the backend to the frontend.
3. **Performance Tuning:**
   * **Optimization**: Fine-tune your API and frontend code for speed and reliability.

**Total Time Allocation**: 2-3 weeks

## Concepts:

### Software Testing Methods

* + **What to Study**: Different methods and approaches to software testing.
  + **Time Needed**: 5 days
  + **Resources**:
    - [Software Testing Tutorial](https://www.guru99.com/software-testing.html)

### Performance Tuning

* + **What to Study**: Techniques to optimize web performance.
  + **Time Needed**: 4 days
  + **Resources**:
    - [Web Performance Optimization](https://developers.google.com/web/fundamentals/performance/why-performance-matters)

## Facts:

### Importance of End-to-End Testing

* + **What to Study**: The value of comprehensive testing that simulates a user's experience.
  + **Time Needed**: 2 days
  + **Resources**:
    - [End-to-End Testing Guide](https://www.cypress.io/blog/2019/12/04/the-complete-guide-to-end-to-end-testing-with-cypress/)

### Consequences of Poor API Performance

* + **What to Study**: The impact of sluggish API performance on user experience and system functionality.
  + **Time Needed**: 2 days
  + **Resources**:
    - [API Performance Impact](https://nordicapis.com/5-ways-poor-api-performance-impacts-the-bottom-line/)

## Procedures:

### How to Perform Full-Stack Integration Tests

* + **What to Study**: Steps to effectively conduct full-stack integration testing.
  + **Time Needed**: 4 days
  + **Resources**:
    - [Full-Stack Testing Guide](https://www.fullstacktesting.com/)

### Methods to Fine-Tune Performance

* + **What to Study**: Techniques for optimizing the performance of both frontend and backend components.
  + **Time Needed**: 4 days
  + **Resources**:
    - [Web Performance Best Practices](https://www.imperva.com/blog/web-performance-optimization-best-practices/)

### How to Use Automated Testing Frameworks for Full-Stack Testing

* + **What to Study**: Approaches to automate testing for both frontend and backend components.
  + **Time Needed**: 2 days
  + **Resources**:
    - [Automated Testing Tutorial](https://www.testim.io/blog/test-automation-tutorial-a-guide-to-start-quickly/)

# Phase 6: Deployment & Monitoring

1. **Monitoring & Logging:**
   * **Features**: Real-time application logging, system alerts, and performance metrics.
   * **Tools**: Grafana, Prometheus, Docker.
2. **Deployment:**
   * **Environment**: Docker containerization and Kubernetes for orchestration.
   * **Monitoring**: Grafana or Prometheus for real-time application monitoring.

**Total Time Allocation:** 2-3 weeks

## Concepts:

### Docker Containerization

* **What to Study**: Understanding Docker and how to create and manage containers.
* **Time Needed**: 4 days
* **Resources**:
  + [Docker Docs](https://docs.docker.com/get-started/overview/)

### Deployment Strategies

* **What to Study**: Different deployment strategies and best practices.
* **Time Needed**: 3 days
* **Resources**:
  + [Deployment Strategies Explained](https://www.weave.works/blog/kubernetes-deployment-strategies)

### Monitoring Tools

* **What to Study**: Tools available for monitoring application performance.
* **Time Needed**: 3 days
* **Resources**:
  + [Monitoring Tools Overview](https://stackify.com/best-application-monitoring-tools/)

### Log Management

* **What to Study**: Managing logs for monitoring and debugging.
* **Time Needed**: 2 days
* **Resources**:
  + [Log Management Guide](https://logz.io/learn/complete-guide-elk-stack/)

## Facts:

### Benefits of Containerization

* **What to Study**: Advantages of using containerization in deployment.
* **Time Needed**: 2 days
* **Resources**:
  + [Containerization Benefits](https://www.ibm.com/cloud/learn/containerization)

### Importance of Real-Time Monitoring

* **What to Study**: Why real-time monitoring is essential.
* **Time Needed**: 2 days
* **Resources**:
  + [Importance of Monitoring](https://www.paessler.com/it-explained/monitoring)

## Procedures:

### Steps for Docker Containerization

* **What to Study**: How to create, deploy, and manage Docker containers.
* **Time Needed**: 4 days
* **Resources**:
  + [Docker Containerization Tutorial](https://www.tutorialspoint.com/docker/index.htm)

### How to Deploy Using Kubernetes

* **What to Study**: Steps to deploy applications using Kubernetes.
* **Time Needed**: 4 days
* **Resources**:
  + [Kubernetes Deployment](https://kubernetes.io/docs/tutorials/kubernetes-basics/deploy-app/deploy-intro/)

### Setting Up Grafana or Prometheus for Monitoring

* **What to Study**: Setting up Grafana or Prometheus for real-time application monitoring.
* **Time Needed**: 3 days
* **Resources**:
  + [Grafana Setup](https://grafana.com/docs/grafana/latest/getting-started/getting-started-prometheus/)
  + [Prometheus Setup](https://prometheus.io/docs/prometheus/latest/getting_started/)

# Phase 7: Maintenance & Upgrades

1. **New Features & Updates:**
   * **Features**: Regular software updates, new feature rollouts based on user feedback.
   * **Strategies**: Agile development, A/B testing for new features.

**Total Time Allocation**: Ongoing (Initial setup: 1-2 weeks)

## Concepts:

### Software Maintenance Strategies

* **What to Study**: Different strategies for maintaining and updating software.
* **Time Needed**: 3 days
* **Resources**:
  + [Software Maintenance](https://www.tutorialspoint.com/software_engineering/software_maintenance_overview.htm)

### Agile Development

* **What to Study**: Agile methodologies for software development.
* **Time Needed**: 2 days
* **Resources**:
  + [Agile Basics](https://www.atlassian.com/agile)

### A/B Testing

* **What to Study**: How to conduct A/B testing for evaluating new features.
* **Time Needed**: 2 days
* **Resources**:
  + [A/B Testing Guide](https://www.optimizely.com/optimization-glossary/ab-testing/)

## Facts:

### Importance of Monitoring User Metrics

* **What to Study**: Importance of monitoring user engagement and behavior.
* **Time Needed**: 1 day
* **Resources**:
  + [User Metrics](https://clevertap.com/blog/key-performance-indicators/)

## Procedures:

### How to Perform Regular Updates

* **What to Study**: Steps for performing regular software updates.
* **Time Needed**: 3 days
* **Resources**:
  + [Software Updates](https://www.comparitech.com/antivirus/how-do-software-updates-work/)

### Methods for Rolling Out New Features Based on User Feedback

* **What to Study**: How to implement new features based on user feedback and data.
* **Time Needed**: 3 days
* **Resources**:
  + [Feature Rollout](https://www.productplan.com/glossary/feature-rollout/)