Project Name: PolitiPulse
Team Members: Adam Botens, Austin Edwards

Abstract:

PolitiPulse aims to create a dynamic online platform that provides users with an informed perspective on political candidates and their positions. The project incorporates three main components: candidate information aggregation, a political spectrum alignment tool, and an AI-driven content summarization feature. By leveraging modern web development technologies and advanced AI capabilities, the project seeks to offer users a comprehensive and user-friendly resource for political engagement and decision-making.

Detailed Report:

In an era marked by increasing political engagement and the significance of well-informed electoral choices, there is a growing need for accessible and organized political information. The proposed website seeks to address this need by offering users a centralized hub for exploring candidates' backgrounds, positions, and bills, as well as an interactive tool to align their political leanings and obtain candidate recommendations.

The project's first component involves the aggregation of candidate-related data from diverse sources, including public statements, speeches, co-authored and cosponsored bills, and funding sources. The United States Government stores an up-to-date database of all House, Senate, and passed bills in an easily retrievable XML format for the express purpose of software development. For this data to be used in a variety of ways PolitiPulse will store this data in its own database by designing an algorithm to authenticate the data for errors, use data map techniques to take the data from the XML file to our database, and do a final readover to ensure no errors are found within the new dataset. This comprehensive collection of information will be made easily accessible through an intuitive user interface. By employing web scraping techniques and API integration, the system will ensure that users can conveniently search and access candidate data in one location.

The second component focuses on user engagement and interactivity. A carefully designed political spectrum alignment tool (PSAT) will allow users to answer a series of questions that reflect their stances on key political issues. The PSAT will chart the user's political leanings on two axes. Horizontally the user will be plotted between the economic left and economic right. These are broad terms when talking about economics but can be summed up in the economic left wishes for more government involvement within its economy while the economic right wishes for less government involvement within its economy. An algorithm will then analyze these responses and position users on the political spectrum. Utilizing AI, the system will suggest candidates whose positions align with users'. This suggestion will highlight what a candidate says their stance will be and highlight what their voting history says they are. Additionally, an AI-driven subcomponent will highlight the pros and cons of candidates' bills and funding sources based on users' political alignments.

The third component introduces advanced Natural Language Processing (NLP) technology to the project. A Language Model (LM) will be employed to automatically summarize the extensive candidate information gathered in Component 1, summarize lengthy and complex bills into common language, and give pros and cons based on the user's political alignment determined in Component 2. Users will receive concise, coherent summaries of candidates' profiles, enhancing accessibility and saving time for those seeking quick insights.

The project will employ a range of technologies to achieve its objectives. Backend development will utilize frameworks such as Express.js for Node.js, ensuring efficient data management and retrieval. Frontend development will involve JavaScript libraries like React for interactive and responsive user interfaces. Web scraping tools, APIs, and databases will facilitate data aggregation. Advanced AI capabilities will be harnessed for both the political spectrum alignment tool and content summarization, possibly utilizing OpenAI's GPT-3 for natural language processing.

This entire project will be hosted on the Google Cloud Platform (GCP) due to server scalability, ease of setup for security features due to the team's lack of cyber security expertise, quick Firebase port for mobile users, CloudSQL for quick database, website hosting, and traffic monitoring. The GCP free trial will be terminated on November 14th, in which the team will split the cost, or if the site is up and running for public access, ad revenue will help cover the cost in the midst of the 2024 presidential election. The GCP comes with a built-in development feature allowing real-time code updates across teams as well as a version history for server backups.

PolitiPulse aspires to provide an innovative solution for political engagement and education. By amalgamating candidate data, interactive assessment tools, and AI-driven content summarization, the platform aims to empower users with the information necessary to make informed political choices. This project stands at the intersection of technology and civic responsibility, offering a user-centric approach to political understanding in the digital age.

Feature List:

Completed by EOS:

- User Authentication and Profile
  - User registration and login functionality
  - User profiles where users can view and update their information.
- Candidate Information Aggregation

    - Aggregation of candidates' public statements, and speeches, co-authored bills, cosponsored bills, and funding sources from reliable sources.
    - Search functionality to easily find and access candidate information
    - Filter options to sort candidates based on various criteria.
- Political Spectrum Test and Alignment
    - Political spectrum test with a set of questions to determine users' political alignment.
    - Algorithm to calculate users' political positions based on their test responses.
    - Candidate matching feature that aligns users with candidates sharing similar political positions.
    - AI-driven subcomponent that analyzes candidates' bills and funding sources, highlighting alignment-specific pros and cons.
- LLM Content Summarization
    - Content summarization utilizing a Language Model (LLM) to highlight key points from Component 1 for each candidate.
    - Integration of LLM-generated content summaries into the user interface.
    - Expandable content sections for users to access detailed information if desired.
- Data Visualization and Insights
    - Visual representations of users' political alignment on a spectrum.
    - Graphs or charts illustrating the distribution of users' political positions.
    - Visual summaries of candidates' positions relative to the user's alignment.
- Security and Privacy
    - Secure encryption for user data and communications.
    - Privacy settings to control what information users want to share.
- Basic Analytics
    - Demo Reports
    - Data Export

Completed if Time Allows:

- User Interaction and Engagement
    - Interactive user interfaces for each component provide a seamless user experience.
    - User-friendly navigation with clear menu options and intuitive design.
    - Social sharing options for users to share their results or interesting candidate information.
    - Option for users to save favorite candidates or results for future reference.

- Responsive Design and Accessibility
    - Mobile-responsive design for optimal performance on various devices.
    - Accessibility features to ensure the website is usable by a wide range of users.
- Admin Panel
    - Admin dashboard to manage user accounts, candidate data, and user-generated content.
    - Tools for adding, updating, or removing candidates and their information.

Wish List:

- Testing and Maintenance
    - Regular testing of features, functionalities, and data accuracy.
    - Scheduled maintenance to ensure smooth operation and data updates.
- Analytics and Reporting
    - Analytics tools to track user engagement, popular candidate profiles, and test completion rates.
    - Reporting capabilities to generate insights for improving the website.
- Community Interaction and User-Generated Content
    - User comments and discussions on candidate profiles.
    - User-generated content, such as user-submitted information on candidate positions.
    - Community forums for discussing political topics and sharing insights.

Technology:

- Frameworks - Django, React
- Operating system - Windows
- Database - CloudSQL
- API - RESTful APIs, Whisper API
- Programming Languages - Python, HTML/CSS, JavaScript, PHP
- 3rd Party Libraries - BeautifulSoup, NLTK, Scikit-learn
- LLM - OpenAI's GPT-3
- Server Software - Google Cloud
- Communication Software - Discord

Server information: We will be using Google App Engine for the website's server. This comes with the GCPs free trial until November 14, after which will cost $0.05 an hour of use, given we stay in the instance class of F1.

Data Sources: The primary data source is GovInfo, a comprehensive library that provides XML versions of Senate, House of Representatives, and executive actions for developers to incorporate into their software applications. Additionally, GovInfo houses public speeches delivered by members of Congress and the executive branch. As for financial election records, the FEC maintains a library containing information on each donation, with private citizen donations being the only data protected for privacy. It's important to note that all the material used in this context is in the public domain.

Team members' backgrounds:

Adam: A computer science student with experience in algorithms, data structures, and object-oriented programming. Has worked on projects such as a hand-gesture-controlled drone, a digital image-reproducing robot, a small-business owner tax tracker, and a stock market predictor using various machine learning models. Relevant skills include Python, MySQL, HTML/CSS, Javascript, and Linux, as well as some networking experience.

Austin: A software engineering student with a solid background in data management, manipulation, and analysis. Academic projects include a drone taking commands from hand signals, making a turtlebot draw on a piece of paper from the image a user draws on a custom paint app, and an autonomous 3D path analyzer for picking paths in an Amazon Warehouse. Relevant skills include an introduction to website building with intermediate javascript skills, successful app integration with a preexisting cloud application, intermediate data management and analysis skills, and advanced Python skills.

Dependencies, limitations, and risks:

Some dependencies include the reliance on data sources. The project heavily relies on external data sources for candidate information, such as public statements, bills, and funding sources. Dependencies on these sources being accurate and up-to-date are critical.

Another dependency is integration with third-party APIs for data retrieval and AI services (such as OpenAI's GPT-3) for content summarization. This introduces dependencies on the reliability and availability of these APIs. The same applies to cloud services. We will have to rely on Google Cloud's reliability and performance.

Limitations include data accuracy, user privacy, AI capabilities, and scalability. Concerning data accuracy, the accuracy of candidate information gathered from external sources can be a limitation. Any inaccuracies or bias in the data can affect the website's credibility. Given that the current data sources are report filings, there shouldn't be a big sway in bias to one party or another. User privacy concerns involve handling user data, especially political preferences, which requires careful attention to privacy concerns and compliance with data protection regulations. The effectiveness of the AI-driven content summarization and political alignment tool depends on the quality and training of the AI models used. These models may have limitations in understanding nuanced political contexts. As far as scalability goes, the project may face scalability challenges during high-traffic periods, such as elections or major political events, which could impact website performance.

Risks Include:

- **Data Reliability:** Reliance on external data sources introduces the risk of receiving outdated or inaccurate information, which could mislead users.
- **API Availability:** If third-party APIs used for data retrieval or AI services become unavailable or change their terms, it can disrupt the functionality of the website.
- **Security:** Risks associated with data breaches and security vulnerabilities could compromise user data or the integrity of the website.
- **User Trust:** Inaccurate candidate matches or content summaries generated by AI could erode user trust in the website's recommendations and information.
- **Legal Compliance:** Non-compliance with data privacy and intellectual property laws could lead to legal challenges and liabilities.
- **Competition:** The project may face competition from other political information websites or platforms, impacting its ability to attract and retain users.
- **User Bias:** There's a risk of user bias in political alignment assessments, potentially leading to recommendations that reinforce existing beliefs.

Timeline:

Month 1: Project Initiation and Planning

Week 1-2: Project Kickoff

- Define project scope, objectives, and success criteria.
- Write a project proposal.
- Assemble the project team and assign roles and responsibilities.
- Create a detailed project plan and timeline.

Week 3-4: Research and Data Sources

- Identify and secure reliable data sources for candidate information.
- Start creating data scraping algorithms for getting candidate information
- Research and select external APIs for data mapping and storage in CloudSQL.

Month 2: Development and Backend Setup

Week 5-6: Backend Development

- Set up the development environment (servers, databases).
- Develop the backend logic for user accounts and candidate data management.
- Begin designing the database schema.

Week 7-8: Data Aggregation

- Implement web scraping or API integration for candidate data.
- Develop data processing scripts to clean and organize the gathered information.
- Implement data mapping to database once candidate data has been scraped

Month 3: Frontend and AI Integration

Week 9-10: Frontend Development

- Begin creating user interfaces for all three project components.
- Focus on the user registration and login functionalities.

Week 11-12: Political Spectrum Tool and AI Integration

- Develop the political spectrum assessment tool and algorithm.
- Research the political compass test question bank
- Explore and experiment with AI models for content summarization.

Month 4: Testing, Deployment, and Launch

Week 13: Testing and Quality Assurance

- Conduct comprehensive testing of all project components.
- Address any bugs, usability issues, or security concerns.

Week 14: User Testing and Feedback

- Invite a small group of users to test the website and provide feedback.
- Make necessary adjustments based on user feedback.

Week 15: Deployment and Final Preparations

- Deploy the website on a production server or cloud platform.
- Configure security measures, including encryption and user data protection.
- Conduct load testing to ensure scalability.

- Cleanup and polish UI

Week 16: Launch and Marketing

- Officially launch the website to the public.
- Implement marketing and outreach efforts to attract users.
- Monitor website performance and user engagement.