PolitiPulse

By: Adam Botens and Austin Edwards

Project Name

The project name will be PoltiPulse, which is short for Political Pulse. The project itself will help its users stay in the know for elections by tracking candidates voting trends, campaign promises, financial data, and the users' political leanings. In short, PoltiPulse gives users a quick pulse on the state of the government and the current election.

Team Members Names

Team Member Name	Role
Adam Botens	Developer, Architect
Austin Edwards	Developer, Architect

Abstract

In the modern-day political landscape, it becomes hard to determine a candidate's true values and motives. PoltiPulse seeks to give power back to the people by making information pertaining to the candidates just a few clicks away. With the user taking a test to see where their political affiliations lie on the political compass, PoltiPulse's first goal is to take that and quickly match up potential candidates the user aligns with. PoltiPulse's second goal is to present factual data on the candidate's political career by collecting bills they sponsored and passed, bills they cosponsored, and bills they voted yes and no on. This includes those that did not pass Congress or got vetoed by the president. The third goal of PoltiPulse is a summative feature that takes the bill's language and condenses it down into main readable points to give users a quick rundown of the bill's purpose and effect. In synergy, these components unite to fulfill PoltiPulse's mission: ushering in a new era characterized by an electorate better informed and equipped to make enlightened choices.

Requirements List

1. **(Onboarding) Login Form**:
    1.1. The login form will be linked to from the homepage and will consist of:

1.1.1. A header that says "Login Form"

1.1.2. A label to identify the email input box

1.1.3. An input box for the user's email

    1.1.3.1. There will be a placeholder "johnDoe@email.com"

1.1.4. A label to identify the password input box

1.1.5. An input box for the password

    1.1.5.1. There will be a placeholder "Password"

1.2. A button labeled "Login"

    1.2.1.1. Will be aligned center under the password input box

    1.2.1.2. On click, it will query the account database for matching user credentials

        1.2.1.2.1. If a match is found, the user returns to the homepage

        1.2.1.2.2. If no match is found or the password is incorrect, a display message will appear above the login button

        1.2.1.2.3. Suspicious login attempts (e.g., repeated failed login attempts from the same IP address) will be handled through CAPTCHA challenges provided by Firebase Authentication

    1.2.2. A button under the Login button labeled "Register"

        1.2.2.1. On click, it will open the registration page in the same tab

1.3. A link to Google sign-in

    1.3.1. On click, it will open a popup to Google Auth

2. **(Onboarding) Registration Form**:

2.1. The registration form will be linked to from the login page and will consist of

    2.1.1. A header that says "Registration Form"

    2.1.2. A label to identify the email input box

    2.1.3. An input box for the email

        2.1.3.1. The placeholder in the email box will be "johnDoe@email.com

        2.1.3.2. The input box will accept only valid emails

            2.1.3.2.1. Emails must contain an @ symbol

            2.1.3.2.2. Validation will be done using HTML input attributes

            2.1.3.2.3. On invalid entries, a display message will appear above email label

    2.1.4. A label to identify the username input box

    2.1.5. An input box for the username

        2.1.5.1. The placeholder in the username box will be JohnDoe

        2.1.5.2. The input box will only accept valid usernames

            2.1.5.2.1. Must contain at least one capital letter

2.1.5.2.2.   Must contain at least one undercase letter

2.1.5.2.3.   Must contain at least two numbers (0-9)

2.1.5.2.4.   Must contain at least one special character (!, @, #, $, %, ^, &, *)

2.1.5.2.5.   Must be at least eight characters long

2.1.6.   A label to identify the password input box

2.1.7.   An input box for the password

    2.1.7.1.   Passwords must be at least eight characters long

    2.1.7.2.   Passwords must contain a special character (!, @, #, $, %, ^, &, *)

    2.1.7.3.   Passwords must contain one capital letter

    2.1.7.4.   On invalid entries, a display message will appear above email label

2.1.8.   A label to identify the confirm password input box

2.1.9.   An input box to confirm the password

    2.1.9.1.   The input box will have "Password" as a placeholder

    2.1.9.2.   The input box will accept only valid passwords

        2.1.9.2.1.   The input must match the input from the first password insertion

        2.1.9.2.2.   On invalid entries, a display message will appear above email label

2.1.10.   A label to identify the State dropdown menu

2.1.11.   A dropdown menu for the State Dropdown menu

    2.1.11.1.   Lists all 50 states in their aberrated forms

    2.1.11.2.   Once the User chooses a state, opens Congressional District Menu

2.1.12.   A label to identify the Congressional District Menu

2.1.13.   A dropdown menu for the Congressional District Menu

    2.1.13.1.   Lists all Congressional Districts based on State Choice.

2.1.14.   A label to identify the Birthday Year box

2.1.15.   An input box to confirm the Birthday Year

    2.1.15.1.   The input box will have "XXXX" as a placeholder

    2.1.15.2.   The input box will accept only valid years (Current Year - 100 to Current Year - 14)

2.1.16.   A label to identify the Birthday Month box

    2.1.16.1.   The input box will have "XX" as a placeholder

    2.1.16.2.   The input box will accept only valid months (01-12)

    2.1.16.3.   Once filled with a valid label, it unlocks the Birthday box.

2.1.17.   A label to identify the Birthday box

2.1.18.   An input box to confirm the Birthday

    2.1.18.1.   The input box will have "XX" as a placeholder

2.1.18.2. The input box will accept only valid days

2.1.18.2.1. Conditional on Month Set

2.1.18.2.1.1. Jan, Mar, May, July August, Oct, Dec = 31

2.1.18.2.1.2. Apr, June, Sep, Nov = 30

2.1.18.2.1.3. Feb = 28 (+1 if leap year)

2.1.19. An input box to confirm the Birthday

2.1.20. A header text that says "Finish"

2.1.21. A button labeled "Register"

2.1.21.1. It will be aligned bottom-right from the confirm password input box

2.1.21.2. On click, it will query Firebase Authentication

2.1.21.2.1. If an email match is found, it will display a message stating that the email is already in use

2.1.21.2.2. If a username match is found, will display a message stating that the username is already in use

2.1.21.3. If no match is found

2.1.21.3.1. The two passwords are compared

2.1.21.3.1.1. If they are equal

2.1.21.3.1.1.1. The user account is created, logged in, and returned to the homepage

2.1.21.3.1.1.2. Email, Username, Password, State, Congressional District, Birthday Year, Month, and Day are logged in the Account Info Database.

2.1.21.3.1.1.3. Auto transfer back to the login page.

2.1.21.3.1.2. If they are not equal

2.1.21.3.1.2.1. A message will display stating that the passwords do not match

2.1.22. A bordered label with a link back to the login page

2.1.22.1. On click, it will open the login page in the same tab

3. (Navigation) Header

3.1. A button with the label "Home"

3.1.1. Takes the user back to the homepage

3.2. A drop-down menu with the label "House of Representatives"

3.2.1. A menu item with the label "Currently Elected"

3.2.1.1. Takes them to "Current House of Representatives" page

3.2.2. A menu item with the label "House Election"

3.2.2.1. Takes them to the "House Election" page

        3.2.3.   A menu item with the label "Upcoming Business"

            3.2.3.1.   Takes them to the "House of Representatives Upcoming Business" page

        3.2.4.   A menu item with the label "Recent Business"

            3.2.4.1.   Takes them to the "House of Representatives Recent Business" page

    3.3.   A dropdown menu with the label "Senate"

        3.3.1.   A menu item with the label "Currently Elected"

            3.3.1.1.   Takes them to "Current Senate" page

        3.3.2.   A menu item with the label "Senate Election"

            3.3.2.1.   Takes them to the "Senate Election" page

        3.3.3.   A menu item with the label "Senate Business"

            3.3.3.1.   Takes them to the "Senate Upcoming Business" page

        3.3.4.   A menu item with the label "Recent Business"

            3.3.4.1.   Takes them to the "Senate Recent Business" page

    3.4.   A dropdown menu with the label "User"

        3.4.1.   A menu item with the label "User Profile"

            3.4.1.1.   Takes them to the "User Profile" page

        3.4.2.   A menu item with the label "Political Spectrum"

            3.4.2.1.   Checks if they have taken the Political Spectrum

                3.4.2.1.1.   If they have not

                    3.4.2.1.1.1.   Takes them to the "Political Spectrum Questionnaire" page

                3.4.2.1.2.   If they have

                    3.4.2.1.2.1.   Dialog box Asks if they want to retake it

                        3.4.2.1.2.1.1.   If the user clicks yes

                            3.4.2.1.2.1.1.1.   Takes them to the "Political Spectrum Questionnaire" page

                        3.4.2.1.2.1.2.   If the user clicks no

                            3.4.2.1.2.1.2.1.   Takes them to "User Info" page

    3.5.   A button with the label "About"

        3.5.1.   Takes them to the "About" page

    3.6.   A button with the label "Contact"

        3.6.1.   Takes them to the "Contact" page

4. **(WP) House of Representatives - Current Elected**

    4.1.   Drop down list to sort them by state

        4.1.1.   Each representative will have a sub-page with a list of the most recent actions and most important bills.

11. (WP) Senate - Recent Business

    11.1.   Show an image list of all bills voted upon for the last 3 sessions.

        11.1.1.   A section for passed bills

        11.1.2.   A section for rejected bills

12. (UI) Bills

    12.1.   Bills will be displayed in image + title button

        12.1.1.   Once clicked, two versions of the bill will come up on screen

            12.1.1.1.   One will be the original bill

            12.1.1.2.   Second will be the summary bill generated in plain English by the LLM

        12.1.2.   Each bill will be generally charted onto the political compass based upon its contents.

13. (PS) Political Spectrum Questionnaire

    13.1.   A paragraph explaining that upon clicking a link, the user will be redirected to https://www.politicalcompass.org/ to take a political spectrum questionnaire. The user will be requested to input their results below.

        13.1.1.   A button labeled "Political Compass"

            13.1.1.1.   On click will redirect the user to https://www.politicalcompass.org/

        13.1.2.   A label that says "Economic Left/Right"

            13.1.2.1.   An input box next to the label with placeholder "0.00"

        13.1.3.   A label that says "Social Libertarian/Authoritarian

            13.1.3.1.   An input box next to the label with placeholder "0.00"

        13.1.4.   A button that says "Submit"

            13.1.4.1.   Upon clicking will send information to the user's profile information in the database

14. (WP) Contact Page

    14.1.   Will be linked to from the questionnaire and from the website navigation panel

        14.1.1.   A label that states "Contact Us"

        14.1.2.

        14.1.3.   A label that identifies the "Name" input

        14.1.4.   An input box for user's name

            14.1.4.1.   The placeholder text will be "john"

        14.1.5.   A label that identifies the "Email Address" input

        14.1.6.   An input box for the user's email

            14.1.6.1.   The placeholder in the email box will be "johnDoe@email.com

            14.1.6.2.   The input box will accept only valid emails

14.1.6.2.1. Emails must contain an @ symbol

14.1.6.2.2. Validation will be done using HTML input attributes

14.1.6.2.3. On invalid entries, display message will appear above email label

14.1.7. A label that identifies the "Subject" input

14.1.8. An input box for the subject

14.1.8.1. Must not be longer than 25 characters

14.1.9. A label that identifies the "Message input

14.1.10. An input box for the message

14.1.10.1. Must not be longer than 500 characters

14.1.11. A CAPTCHA/ReCAPTCHA will be under the final input box

14.1.11.1. Google Recaptcha provided by Google

14.1.12. A submit button in the bottom right

14.1.12.1. Upon clicking:

14.1.12.1.1. Will send to a shared email designated for use of the website

15.   (WP) User Profile

15.1. Linked to from the website's navigation panel

15.1.1. A header that says "User Profile"

15.1.2. A form

15.1.2.1. A header that says "Political Spectrum"

15.1.2.1.1. A label that states "General: "

15.1.2.1.2. A label that displays the user's general political "title"

15.1.2.1.3. A label that states "Social"

15.1.2.1.4. A label that displays the user's political score on social policies

15.1.2.1.5. A label that states "Economic"

15.1.2.1.6. A label that displays the user's political score on economic policies

15.1.2.1.7.

15.1.2.2. A header that says "User Details"

15.1.2.2.1. A label identifying the user's username

15.1.2.2.2. A label identifying the user's email address

15.1.2.2.3. A label identifying

15.1.2.3. A header that says "Personal Information:"

15.1.2.3.1. A label identifying the user's name

15.1.2.3.2. A label identifying the user's location

15.1.2.3.3. A label identifying the user's congressional district

15.1.2.3.4.   A label identifying the user's birthday

15.1.2.4.   A header that says "Password Change"

15.1.2.4.1.   A label to identify the current password input

15.1.2.4.2.   An input box for the current password

15.1.2.4.2.1.   There will be a placeholder "Current Password"

15.1.2.4.3.   A label to identify the new password input box

15.1.2.4.4.   An input box for the new password

15.1.2.4.4.1.   Passwords must be at least 8 characters long

15.1.2.4.4.2.   Passwords must contain a special character (!, @, #, $, %, ^, &, *)

15.1.2.4.4.3.   Passwords must contain one capital letter

15.1.2.4.5.   A label to identify the confirm password input box

15.1.2.4.6.   An input box to confirm the new password

15.1.2.4.6.1.   Passwords must be at least 8 characters long

15.1.2.4.6.2.   Passwords must contain a special character (!, @, #, $, %, ^, &, *)

15.1.2.4.6.3.   Passwords must contain one capital letter

15.1.2.4.6.4.   Password must match the first password that was input

15.1.2.4.7.   A submit button labeled "Submit"

15.1.2.4.7.1.   On click:

15.1.2.4.7.1.1.   If current password is correct, and both new passwords match

15.1.2.4.7.1.1.1.   Change user's password and reload profile page

15.1.2.4.7.1.2.   On invalid entries

15.1.2.4.7.1.2.1.   Display message will appear below submit button

15.1.2.5.   A header that says "Actions"

15.1.2.5.1.   A button that says "Delete Profile"

15.1.2.5.2.   Upon clicking brings up a modal

15.1.2.5.2.1.   Modal label reads "Delete Account?"

15.1.2.5.2.2.   An OK button in bottom right

15.1.2.5.2.2.1.   Removes user's account

15.1.2.5.2.3.   A Cancel button on left of OK button

15.1.2.5.2.3.1.   Closes the modal

16.   (WP) About Page

16.1.   Linked to from the website navigation panel

16.1.1. A header that says "About Us"

16.1.2. A paragraph that explains the website

16.1.3. A paragraph that talks about Adam

16.1.4. A paragraph that talks about Austin

17. (DB) Database - User Account

17.1. Stores user information

17.1.1. Fields

17.1.1.1. User ID (Primary Key)

17.1.1.2. Username (Unique)

17.1.1.3. Password

17.1.1.3.1. Salting the password before passing to hash

17.1.1.3.2. SHA-256 Hash for password security

17.1.1.3.3. Store hashed password.

17.1.1.4. Email

17.1.1.5. Birthday

17.1.1.6. State

17.1.1.7. Congressional District

17.1.1.8. xAxis score

17.1.1.9. yAxis score

17.1.1.10. General alignment

17.1.1.11. Last Successful Login Timestamp

18. (DB) Database - Candidates

18.1. Contains information about each political candidate

18.1.1. Fields

18.1.1.1. Candidate ID (Primary Key)

18.1.1.2. Name

18.1.1.3. Profile Picture URL

18.1.1.4. Brief Description

18.1.1.5. Party Affiliation

18.1.1.6. Overall political Alignment (to match with user results)

18.1.1.7. District

18.1.1.8. State

19. (DB) Database - Bills

19.1. Contains information about various bills

19.1.1. Fields

19.1.1.1. Bill ID (Primary Key)

19.1.1.2. Bill Name

19.1.1.3. Brief Description

19.1.1.4.   Full Bill Text

19.1.1.5.   Date Introduced

19.1.1.6.   Status (Passed, Vetoed, KIC (Killed in Committee)

19.1.1.7.   Generated Summary

20.   (DB) Database - Candidate Bill Actions

20.1.   Links candidates to their actions related to specific bills

20.1.1.   Fields

20.1.1.1.   Action ID (Primary Key)

20.1.1.2.   Candidate ID (Foreign Key Linking to Candidates Database)

20.1.1.3.   Bill ID (Foreign Key linking to bills database)

20.1.1.4.   Action Type (Sponsored, Co-Sponsored, Voted Yes, Voted No)

21.   (BE) Bill Summarization Algorithm

21.1.   Takes the original bill text

21.1.1.   Runs it through a LLM and shortens it to its main points

21.1.2.   Runs it through again translated the legal speak into plain english

21.1.3.   Stores summarized bill into Bills Database - Generated Summary

PolitiPulse Tech-Stack

Place In System	Dependency Name	Dependency Type	Why
Frontend	React	JS Library	Building dynamic and interactive user interfaces. Scalability and maintainability.
Frontend	Replit	Hosting	Cheap web hosting
Frontend	Firebase Authentication	Firebase Module	For managing the application's state, including user authentication status and candidate-related data.

Frontend	Chart.js	JS Library	Creating interactive and dynamic data-driven visualizations. Will be used for displaying data.
Frontend	Firebase Platform	Cloud Platform	For handling frontend -> backend and providing authentication and storage
Backend	Node.js	Runtime Environment	Will be used for building the backend
Backend	Express.js	Framework	Will be used for building the server
Backend	Firestore Database	Database	NoSQL database for user info and aggregated data
Backend	GovInfo API	API	For retrieving data from GovInfo

Tentative Schedule

Week 1-3: Planning Phase

Week 1

Define the project's full scope and deliverables.
Conduct thorough research on the political spectrum, bill data sources, and other required components.
Start drafting the software design, data models, and preliminary UI/UX designs.

Week 2

Continue with data model finalization.
Detailed planning of database structures and relations.
Initial setup of the development environment, including databases.

Week 3

Finalize UI/UX designs for core components like Login, Registration, Header, and Bill UI.
Plan sprints, tasks, and prioritize components based on dependencies.

Week 4

Finalize UI/UX designs for core components like Login, Registration, Header, and Bill UI.
Plan sprints, tasks, and prioritize components based on dependencies.

Week 5-7: Initial Development

Week 5

Programmer A: Begin Login & Registration backend development, focusing on the User Account and IP Address Log databases.
Programmer B: Develop frontend for Login, Registration, Header, and initial structure for the main pages (House of Reps, Senate, etc.)

Week 6

Programmer A: Set up the Political Compass Test database and begin backend for Political Spectrum Questionnaire.
Programmer B: Design and develop the frontend for the Political Spectrum Questionnaire.

Week 7

Both programmers collaborate on integrating components and ensuring the core functionality is smooth.

Week 8-11: Core Features Development

Week 8

Programmer A: Develop backend for House of Representatives & Senate (Current Elected, House Election, Upcoming Business, etc.).
Programmer B: Work on frontend features corresponding to Programmer A's tasks.

Week 9

Programmer A: Develop backend for Bill UI and set up the Bills database and Candidate Bill Actions database.
Programmer B: Construct frontend UI for Bills and integrate with existing features.

Week 10

Programmer A: Set up and develop backend functionality for User Profile and Political Alignment Database.
Programmer B: Work on frontend features for User Profile, Contact Page, and About Page.

Week 11

Both programmers integrate existing components and ensure the flow across the platform is consistent.

Week 12-14: Testing and Refinement

Week 12

Both programmers test all developed features, focusing on user experience, data accuracy, and potential security vulnerabilities.

Week 13

Address identified issues, bugs, and refine features based on feedback.

Week 14

Programmer A: Develop the Bill Summarization Algorithm.

Programmer B: Implement the UI that presents the summarized bills and link it to the Bills Database.

Week 15-16: Finalization and Deployment

Week 15

Review all features, ensure alignment with initial requirements.
Begin preparations for deployment.

Week 16

Deployment preparation: Hosting setup, final checks, documentation preparation, and security audits.

Week 16

Launch the project.
Monitor for issues, feedback, and ensure smooth user experience.