⍰ Fix Policyholder Registration and Login (Not Working)

- Assigned to: Kutyla
- Description: Ensure policyholder registration and login functionalities work seamlessly.
- Plan:
    - Debug frontend form validation, backend API endpoints, and database connectivity.
    - Check for issues like duplicate emails, incorrect passwords, or server errors.
    - Test with sample users across browsers (Chrome, Firefox, Safari).
    - Implement clear error messages (e.g., “Invalid credentials”).
- Priority: High
- Dependencies: None

⍰ Direct to Admin Without Radio Buttons

- Assigned to: Buks
- Description: Redirect users to the admin dashboard without requiring radio button selection.
- Plan:
    - Update routing logic (frontend or backend) to bypass radio button screen.
    - Ensure only authorized admins are redirected using authentication checks.
    - Remove or hide radio button UI elements.
    - Test redirects for different user roles.
- Priority: Medium
- Dependencies: Task 1

⍰ Implement Search Filter in Admin (Filter 4 Main Things)

- Assigned to: Maite
- Description: Add a search filter in the admin dashboard for four key attributes (e.g., policyholder name, ID, policy number, status—confirm with stakeholders).
- Plan:

    - Confirm the four attributes to filter.
    - Create UI components (e.g., text inputs or dropdowns) for filtering.
    - Build backend API endpoints for filter queries (e.g., SQL or NoSQL).
    - Ensure case-insensitive, partial-match support and test with large datasets.
- Priority: Medium
- Dependencies: Task 2

▯ Develop Backend for Beneficiary

- Assigned to: Dipuo
- Description: Build or fix backend logic for managing beneficiaries (CRUD operations).
- Plan:
    - Define beneficiary schema (e.g., name, relationship, ID, contact).
    - Create API endpoints for create, read, update, delete operations.
    - Link beneficiaries to policyholders in the database.
    - Validate inputs and test edge cases (e.g., multiple beneficiaries).
- Priority: High
- Dependencies: Task 1

▯ Rearrange Styling in Pages

- Assigned to: Boitshepo
- Description: Enhance visual layout and styling of pages for better user experience.
- Plan:
    - Review UI/UX of all pages (registration, login, admin dashboard).
    - Update CSS (or framework like Bootstrap/Tailwind) for consistency.
    - Ensure responsive design for mobile and desktop.
    - Test across devices and screen sizes.
- Priority: Low
- Dependencies: None

▯ Implement Previous History for Policyholder Actions

- Assigned to: Apfeh
- Description: Track and display policyholder action history (e.g., policy updates, claims, logins).
- Plan:
    - Create a database table/collection for action logs (action type, timestamp, details).
    - Build an API endpoint to fetch history for a policyholder.
    - Develop a frontend component to display history in the admin/policyholder dashboard.
    - Test with sample actions and ensure performance with large logs.
- Priority: Medium
- Dependencies: Task 1

▯ Integrate Home Affairs Database Connectivity

- Assigned to: Nyiko
- Description: Connect to the Home Affairs database for verification or data retrieval.
- Plan:
    - Identify API or database access method for Home Affairs (confirm credentials/permissions).
    - Implement secure API calls or database queries for relevant data (e.g., ID verification).
    - Handle errors (e.g., downtime, invalid data) and ensure data privacy compliance.
    - Test integration with sample data and verify accuracy.
- Priority: High
- Dependencies: Task 1

▯ Send Verification to Insured Person (Not Beneficiary)

- Assigned to: Mashabela

- Description: Ensure verification (e.g., email/SMS) is sent to the insured person, not the beneficiary.
- Plan:
    - Update verification logic to target the insured person’s contact details.
    - Modify backend workflows for sending notifications (e.g., via email API or SMS gateway).
    - Test with sample insured persons to confirm correct recipient.
    - Ensure compliance with data privacy regulations.
- Priority: Medium
- Dependencies: Task 4

▯ Restore All Functions in Admin Pages

- Assigned to: Letho
- Description: Bring back all functionalities in the admin pages (e.g., dashboards, reports, user management).
- Plan:
    - Identify broken or missing admin functions through testing or stakeholder feedback.
    - Debug and restore each function (e.g., update user data, generate reports).
    - Ensure all admin actions are secure and role-based.
    - Test comprehensively across admin workflows.
- Priority: High
- Dependencies: Task 2