Introduction

This project aims to streamline the management of specific referral asks within a professional group, such as BNI members, using Google Forms, Google Sheets, Apps Script, and Data Studio. The goal is to create an automated and centralized system where members can submit, track, and view referrals efficiently. By leveraging Google Workspace tools, this workflow eliminates manual intervention, enhances visibility, and ensures seamless collaboration among members.

Scope of Work

1. Google Form Development

- **Objective**: Create a simple and user-friendly Google Form for members to submit their specific referral asks.
- **Fields to Include**:
    - **Name**: Member’s full name.
    - **Specific Ask**: Description of the referral they seek.
    - **Date**: Optional date picker for deadline or reference.
- **Outcome**: Submissions will be stored directly in a linked Google Sheet.

---

2. Google Sheet Setup

- **Objective**: Collect, organize, and manage data submitted through the Google Form.
- **Tasks**:
    - Design columns to capture all fields from the form (Name, Ask, Date, Status, etc.).
    - Add a **Status** column to track the state of each ask (e.g., "Open," "Closed").
    - Enable **Filter Views** for:
        - Filtering by member name.
        - Filtering by referral status (e.g., Open or Closed).
- **Outcome**: A centralized database to manage and view all referral data.

---

3. Google Apps Script Automation

- **Objective**: Add automation to the system to reduce manual effort.
- **Tasks**:
    - **Status Management**:

        - Add functionality to mark asks as "Closed" by clicking a button in the Google Sheet.
    - **Notification System**:
        - Automate email or WhatsApp notifications for:
            - New asks submitted.
            - Changes to ask status.
    - **Data Validation**:
        - Ensure no duplicate entries for the same member and ask.
- **Outcome**: Automated workflows to enhance efficiency and real-time notifications.

---

4. Google Data Studio Integration

- **Objective**: Create an interactive and visually appealing dashboard for members to view specific asks.
- **Tasks**:
    - Connect the Google Sheet as a data source in Google Data Studio.
    - Design a dashboard with the following features:
        - Filter controls (e.g., by member name or status).
        - Dynamic charts and tables displaying referral data.
    - Publish the dashboard and generate an embed link for integration.
- **Outcome**: An intuitive dashboard for members to interact with referral data in real time.

---

5. HTML Integration

- **Objective**: Create a centralized HTML page to serve as the front-end interface for the system.
- **Tasks**:
    - Embed the Google Form for submitting asks.
    - Embed the Google Data Studio dashboard for viewing asks.
    - Add a dropdown for dynamically loading filter views from the Google Sheet.
- **Outcome**: A seamless user experience for submitting and managing referrals.

Expected Deliverables

1. A functional Google Form linked to a Google Sheet.
2. A well-organized Google Sheet with filter views for easier navigation.
3. Apps Script automation for:
    - Updating ask statuses.
    - Sending notifications.

4. An interactive Data Studio dashboard.
5. A user-friendly HTML page embedding the form and dashboard.

Key Parameters

1. **Security**:
   - Ensure data is accessible only to authorized members.
   - Use Google Form restrictions and Google Sheet permissions.
2. **Scalability**:
   - Design the system to handle over 100 members and their asks efficiently.
3. **Ease of Use**:
   - The system should be intuitive for non-technical users.
4. **Automation**:
   - Minimize manual interventions by leveraging Apps Script and Data Studio.
5. **Real-Time Updates**:
   - Ensure referral data is always up-to-date and available for members.

Algorithm

```
1. Start
2. Create a Google Form with required fields (Name, Ask, Date).
3. Link the Google Form to a Google Sheet for data collection.
4. In the Google Sheet:
   a. Create filter views for each member or status.
   b. Save and publish the sheet with filters enabled.
5. Write Google Apps Script for:
   a. Automating status updates (Open → Closed).
   b. Sending notifications on form submissions.
6. Open Google Data Studio:
   a. Connect the published Google Sheet as the data source.
   b. Design an interactive dashboard with filter controls.
7. Publish the Data Studio dashboard.
8. Create an HTML page:
   a. Embed the dashboard link in an iframe.
   b. Add a dropdown to load specific filter views dynamically (optional).
9. Test the workflow end-to-end for functionality.
10. End
```

Workflow

1. **Create Google Form**:
   - Design a form with fields like **Name**, **Specific Ask**, and **Date**.
2. **Link Form to Google Sheet**:
   - Set up a linked Google Sheet to automatically collect form responses.
3. **Set Up Filter Views in Google Sheets**:
   - Create filter views for each member or specific criteria like "Open Asks."
4. **Publish Sheet with Filters**:
   - Publish the Google Sheet with its filter views to make it accessible via unique URLs.
5. **Add Google Apps Script for Automation**:
   - Write scripts to:
     - Automatically update the status of asks (e.g., "Open" to "Closed").
     - Send email/WhatsApp notifications on new submissions.
6. **Connect Sheet to Google Data Studio**:
   - Use Data Studio to link the Google Sheet and create an interactive dashboard.
7. **Create Interactive Dashboard**:
   - Add filter controls (e.g., dropdown for member names).
   - Display asks dynamically based on selected filters.
8. **Embed Dashboard in HTML Page**:
   - Copy the embed link from Data Studio and add it to an HTML iframe.

Step 1: Add Google Sheets Filters for Specific Members or Asks

To allow filtering in the Google Sheet:

1. **Set Up Filters**:
   - Open the Google Sheet linked to the form.
   - Highlight the data range (e.g., all rows and columns of your data).
   - Click on **Data > Create a Filter**.
   - This will allow users to filter by member name, ask status, or date.
2. **Publish with Filters Enabled**:
   - Click on **File > Share > Publish to the Web**.
   - Copy the embed link and ensure that the filters are preserved.
   - This allows users to interact with filters directly when viewing the sheet.

Step 2: Use Google Apps Script for Automation

To make the system more automated (e.g., updating statuses, sending notifications):

1. **Open Apps Script**:
   - In your Google Sheet, click on **Extensions > Apps Script**.

2. **Automate Closing of Asks**:
    - Create a script to add a clickable link/button in the sheet to mark an ask as closed.

3. Send Notifications via Email or WhatsApp:

- Use Google Apps Script to send notifications when an ask is submitted or closed.