## **1. Product Overview**

**Name**: Bedieningen Profiel (Five-Fold Ministry Profile)  
 **Purpose**:

* Help church members discover their primary gifting among the Five-Fold Ministries (Apostle, Prophet, Evangelist, Pastor/Herder, Teacher).
* Provide church leaders a dashboard to see both individual and team-level results.
* Offer insight into how these ministry roles are distributed across the congregation or leadership teams.

**Objectives**:

1. **Individual Insights**: Each participant completes a 40-question (or 70-question) survey—14 questions per role, though some roles may have fewer if the total is 40. The app generates an immediate, personalized result indicating their highest-scoring ministry gifting(s).
2. **Team/Church Overview**: Leaders see summarized data in a dashboard—bar charts, pie charts, and tables showing the distribution of scores for the entire group.
3. **Coaching & Follow-Up**: The tool should support leaders in guiding or coaching people according to their top giftings.

## **2. Key Features**

1. **User Authentication & Profile**
   * **Sign Up / Login**: Users register with email/password or via a single sign-on (if needed).
   * **User Profile**: Store name, email, church (or team) affiliation.
2. **Questionnaire / Survey**
   * **Dynamic Survey**: 40 total questions (or 14 per role × 5 roles = up to 70 if needed).
   * **Rating Scale**: Each question has a Likert-type scale from 0–5 (0 = strongly disagree, 5 = strongly agree), or the exact scale you provided.
   * **UX Requirement**: Visually consistent with your design (see attached screenshot), with a simple slider or clickable scale for each statement.
3. **Scoring**
   * **Calculation**: For each role, sum up the answers to that role’s questions (max 5 points each).  
     + E.g. if Role A has 14 questions, max score = 14 × 5 = 70.
   * **Data Storage**: Save raw question-by-question responses in the database, plus the total score for each role.
4. **Dashboard – Individual View**
   * **Summary Graphs**: Bar chart or radar chart showing a user’s five scores (A, P, E, H, L).
   * **Interpretation**: Text or interpretive statement about the user’s top gifting(s).
   * **Possibility to Print or Export**: Option to download or print results as PDF (optional feature).
5. **Dashboard – Team Leader / Admin View**
   * **Team Overview**: Display total scores aggregated across all participants for each role (e.g., a bar chart with aggregated data: “Apostle=140 points,” “Prophet=135,” etc.).
   * **Percentage Breakdown**: Pie chart or donut chart showing the percentage each role occupies within the team total.
   * **Individual Breakdown**: Ability to view individual members’ top roles or entire profile.
   * **Filters**: By small group, ministry team, or other attributes if needed (optional advanced feature).
6. **Design / User Interface**
   * **Color Palette**: A gradient background from pastel green to teal, matching the screenshot.
   * **Layout**: Large, bold headings (e.g., “Inzicht in je gemeenteleden & kerk”), spaced-out sections, modern web fonts.
   * **Responsive**: Must function well on desktop, tablet, and mobile screens.
7. **Administration & Configuration**
   * **Manage Questions**: Admin can add/edit questions or statements for each role if needed (optional advanced feature).
   * **Manage Roles**: Optionally allow admin to adjust role definitions or weighting (optional advanced feature).
   * **User Management**: Admin can view registered users, reset passwords, or remove users if necessary.
8. **Security & Privacy**
   * **Data Security**: All user data should be stored securely with standard encryption.
   * **Privacy Compliance**: Provide a clear privacy policy. Possibly comply with GDPR if you operate in the EU.
9. **Deployment & Hosting**
   * **Replit**: The entire application should run on Replit or a container-based environment, with a front-end and back-end.
   * **Domain**: Possibly configure a custom domain (e.g., [www.bedieningenprofiel.nl](http://www.bedieningenprofiel.nl)).

## **3. Detailed Functional Requirements**

Below is a more structured breakdown:

### **3.1 User Authentication**

* **Requirement**:  
  + Users can create an account with name, email, and password.
  + Users can log in with a valid email and password.
* **Success Criteria**:  
  + Verified by the presence of a user session/cookie upon login.
  + Passwords are securely hashed and not stored in plaintext.

### **3.2 Profile Setup**

* **Requirement**:  
  + After sign-up, the user is prompted to begin the questionnaire.
  + User can optionally fill in additional data (church name, ministry position, etc.).

### **3.3 Questionnaire**

* **Requirement**:  
  + 40 questions total (each mapped to a role). Alternatively, up to 14 questions per role if using a 70-question format.
  + Each question has a statement, a 0–5 slider, or a 6-choice radio button.
  + The question page design should match the screenshot style: statement #1 on the left, statement #2 on the right (if using side-by-side approach).
* **Success Criteria**:  
  + The user can navigate through all questions.
  + The system records each response as a numeric score (0–5).

### **3.4 Scoring Logic**

* **Requirement**:  
  + Each question belongs to exactly one role.
  + The user’s total score for that role is the sum of all question scores in that role.
  + Example: If the user answered the role “Apostle” questions with a total of 55 points out of 70 possible, that is their “Apostle” score.
* **Success Criteria**:  
  + The final numeric score per role is stored in a database table (e.g., user\_scores).

### **3.5 Results / Profile Page**

* **Requirement**:  
  + Show the user’s five role scores with a bar chart or spider/radar chart.
  + Possibly highlight the top 1 or 2 roles with a descriptive label (“Primary gifting: Prophet,” etc.).
  + Provide some text-based interpretation: “Your highest score is in the Prophetic role, which typically means…”
* **Success Criteria**:  
  + The result appears immediately after the user finishes the test.
  + The user can revisit their results any time via a “My Profile” page.

### **3.6 Team/Church Dashboard (for Admin / Pastor)**

* **Requirement**:  
  + Display aggregated scores across all users or across a filtered group (like a small group).
  + Show numeric totals for each role (e.g., “Total Apostle = 140”).
  + Show a bar chart (quantitative) and pie chart (percentage) for each role’s proportion.
  + List top scorers for each role or an overview table with all participants.
* **Success Criteria**:  
  + Admin sees the entire group’s data.
  + Admin can drill down to an individual or view a high-level summary.

### **3.7 Data Export (Optional / Future)**

* **Requirement**:  
  + Admin can export the data to CSV or Excel.
* **Success Criteria**:  
  + The file downloads with columns for user info, role scores, etc.

### **3.8 Notifications (Optional / Future)**

* **Requirement**:  
  + Email the user a summary of their results.
* **Success Criteria**:  
  + Automated email triggered upon survey completion.

## **4. Non-Functional Requirements**

1. **Performance**
   * The system must handle up to N simultaneous users (define a rough estimate).
   * Page loads for the question survey should be near-instant.
2. **Security**
   * Use HTTPS.
   * Store passwords securely.
   * Restrict the admin dashboard behind proper role-based access control.
3. **Scalability**
   * As user growth increases, the solution can scale (Replit can handle additional traffic or DB calls as needed).
4. **Maintainability**
   * Clear separation of front-end and back-end code.
   * Documentation explaining how to add/edit questions and manage roles.
5. **Usability**
   * Consistent UI design.
   * Straightforward instructions on each page so users of all tech skill levels can respond to the questionnaire.

## **5. Technical Stack Recommendations**

* **Front-End**:  
  + Framework: React, Vue, or Next.js (Replit can host a React or Next.js app easily).
  + Charting Library: Chart.js or Recharts for the bar and pie charts.
* **Back-End**:  
  + Node.js / Express (common choice on Replit).
  + Python Flask or Django could also be used, but Node/Express might fit more smoothly if you’re using React.
* **Database**:  
  + SQLite for simplicity on Replit, or a hosted PostgreSQL/MongoDB if you need more robust data handling.
* **Deployment**:  
  + Hosted on Replit with a public URL.
  + Custom domain mapping if desired.

## **6. Milestones & Development Plan**

1. **MVP (Minimum Viable Product) – 2–3 weeks**
   * Basic user registration & login.
   * Single-page test with 40 questions.
   * Score calculation & basic results page with user’s five role scores.
2. **Enhanced Dashboard & Design – 2–4 weeks**
   * Implement the final look & feel from the provided design (green/teal theme).
   * Add charting for individual results (bar chart).
   * Create admin dashboard with aggregated data.
   * Provide bar and pie charts for the team’s scores.
3. **Future Features**
   * Email summaries to users.
   * CSV export.
   * Editing / adding questions from an admin panel.
   * Additional coaching resources or role-based tips.

## **7. Risks & Assumptions**

* **User Experience**: Assume a wide range of tech-savviness. Keep interface intuitive.
* **Data Privacy**: Church members might be sensitive about personal data. Clarify how their scores are used.
* **Question Ownership**: If you have custom questions, ensure they’re validated and well-structured before coding.

## **8. Success Criteria / Acceptance**

1. **Successful Completion of the Survey**
   * Users can answer all questions smoothly and see their results without errors.
2. **Accurate Role Scoring**
   * The final total for each role must align with the user’s input (0–70 for each role, if using 14 questions × 5 points).
3. **Intuitive Admin Dashboard**
   * Admin can easily see and interpret group-level data (scores across all roles).
   * Charts and tables display consistent, correct information.
4. **Positive Feedback**
   * Beta testers or initial church members find the tool valuable and easy to use.