**Racer-One Backlog**

**Sprint 1: Setup and Basic Functionality (Weeks 1-2)**

1. Set up project repository and version control system. (Difficulty: 1)

2. Initialize React project structure. (Difficulty: 1)

3. Create LoginPage component with basic layout and styling. (Difficulty: 2)

4. Implement login functionality to authenticate users. (Difficulty: 3)

5. Develop EmployeeDemographicsPage component layout. (Difficulty: 2)

6. Integrate employee image and basic demographic data display. (Difficulty: 3)

7. Design and implement database schema for storing employee data. (Difficulty: 4)

**Sprint 2: Data Entry and Evaluation (Weeks 3-4)**

1. Design ResumePage component for entering employee resume data. (Difficulty: 3)

2. Develop form fields for capturing various resume data points. (Difficulty: 3)

3. Implement validation for resume form inputs. (Difficulty: 2)

4. Create Evaluation1Page component for inputting performance evaluation data. (Difficulty: 3)

5. Design UI elements for inputting performance ratings and feedback. (Difficulty: 2)

6. Develop Evaluation2Page for conducting spiritual gifts tests. (Difficulty: 3)

**Sprint 3: Team Management and Analysis (Weeks 5-6)**

1. Design TeamRequirementsPage for specifying project requirements. (Difficulty: 3)

2. Implement form fields for defining project goals and team composition. (Difficulty: 3)

3. Develop TeamProjectPage for tracking current projects and teams. (Difficulty: 3)

4. Create AnalyzerPage component for team analysis and recommendation. (Difficulty: 4)

5. Implement logic for calculating similarity scores using the Jaccard Similarity Equation. (Difficulty: 4)

**Sprint 4: Additional Features and Refinement (Weeks 7-8)**

1. Implement AboutPage component for providing app information and details. (Difficulty: 2)

2. Design landing page layout and content. (Difficulty: 2)

3. Add navigation links between different pages. (Difficulty: 2)

4. Refactor codebase for improved readability and maintainability. (Difficulty: 3)

5. Enhance user interface with styling and visual improvements. (Difficulty: 3)

6. Write unit tests for new and existing features. (Difficulty: 3)