**Step 1: Data Retrieval**

1.1. Obtain GitHub API access: Register your application with GitHub to obtain the necessary credentials for API access.

1.2. Authenticate with GitHub: Implement an authentication mechanism to allow users to connect their GitHub accounts securely.

1.3. Retrieve user data: Use the GitHub API to fetch the candidate's profile information, repositories, contributions, skills, and activity.

**Step 2: Data Processing**

2.1. Clean and preprocess data: Perform data cleaning operations to handle missing values, remove duplicates, and address any inconsistencies.

2.2. Normalize data: Standardize and normalize the retrieved data to ensure consistency in format and representation.

2.3. Extract relevant features: Extract key information from the retrieved data, such as repository names, descriptions, programming languages, contributions, and skills.

**Step 3: Report Generation**

3.1. Design the dashboard: Create a user-friendly and visually appealing dashboard interface that presents the candidate's GitHub insights.

3.2. Data visualization: Utilize appropriate data visualization techniques, such as graphs, charts, and tables, to display the information in an intuitive manner.

3.3. Interactive features: Implement interactive features within the dashboard to allow recruiters to explore and interact with the candidate's GitHub profile.

**Step 4: Testing and Quality Assurance**

4.1. Perform unit testing: Validate each component and functionality of the system to ensure they function as expected.

4.2. Conduct integration testing: Test the integration between different modules and components to ensure seamless data flow and functionality.

4.3. User acceptance testing: Engage recruiters or potential users to participate in testing and gather feedback to refine and improve the system.

**Step 5: Deployment and Maintenance**

5.1. Deploy the system: Set up the infrastructure and deploy the system on a server or cloud platform for accessibility.

5.2. Monitor and maintain: Regularly monitor the system's performance, security, and data integrity, and address any issues that arise promptly.

5.3. Enhancements and updates: Continuously gather feedback, incorporate user suggestions, and implement improvements to enhance the system's functionality and user experience.