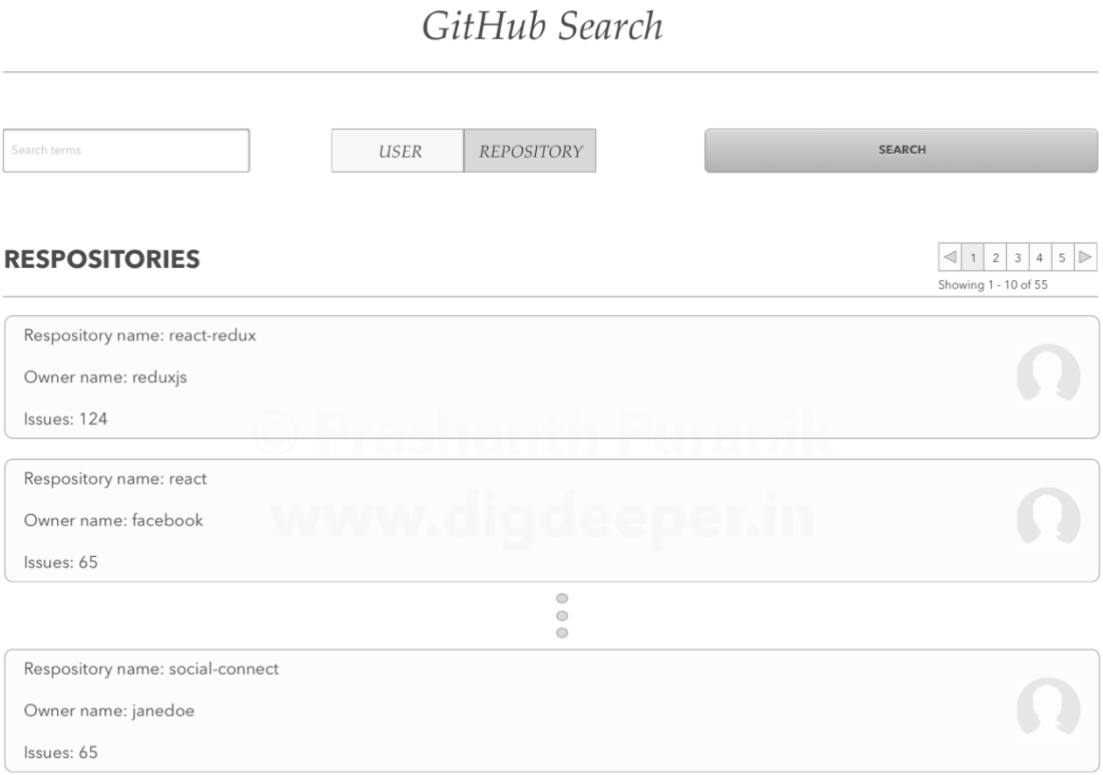
GITHUB SEARCH APP

Use GitHub API (https://developer.github.com/v3/guides/getting-started/) to build a GitHub search app.

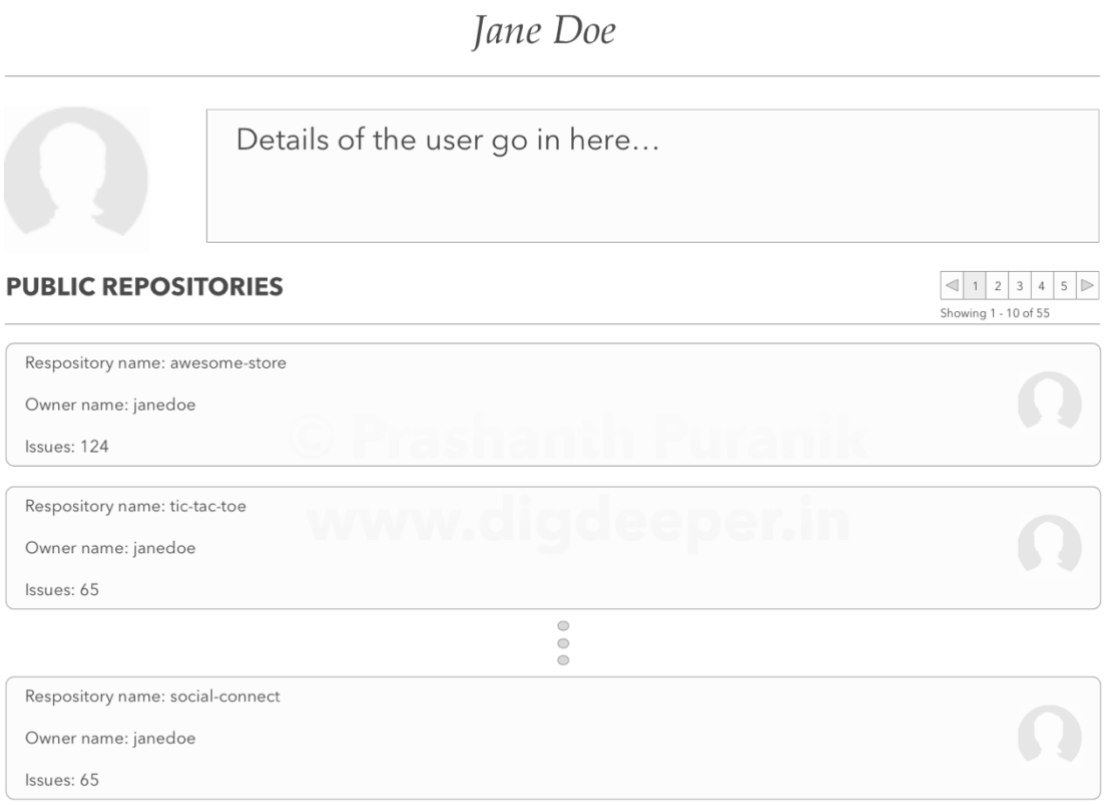
Mock-ups

Page 1a: Search page (User search)

Page 1b: Search page (Repositories search)



Page 2: User details page



Page 3: Repository details page

Functionality

1. The home page has a search box. Additionally, there is a split button USERS | REPOSITORIES. Searching using the search button results in the list of first 10 matched users / repositories getting displayed in a nice list view. Only basic details are given in the list.

The pagination widget can be used to move from one page to the next. The total results are displayed below the pagination widget.

When we click on a user entry in the search result, the details of the user are shown in a user page. When we click on a repository entry in the search result, the details of the user are shown in a repository page.

2. The users details are displayed in this page. It includes the list of public repositories for this user - clicking this should take to the repository page.

3. This displays details of repository. The owner for the project is also provided - clicking this should take one to the user page and show owner details.

Additional Requirements

UI to be designed and developed by your team

You need to come up with your own design of components / pages to **support at least 2 of the following 4 features**. Design user-friendly interfaces. Take care of all different application states (loading, error, success, no data fallback message), responsive web design (phone, desktop), SEO, performance (appropriate rendering model), web a11y. Conduct Lighthouse audit on your app, and check for web vitals.

NOTE: When developing these pages if you feel the need for more pages (like login page, authorization page etc., please develop those as well).

1. **User Activity Feed Page**
   Description: Create a page that displays the recent public events performed by a user, such as pushes, pull requests, and issues.
   - **API Endpoint**: `GET /users/{username}/events/public`
   - **Details to Display**
     - Event type (e.g., PushEvent, PullRequestEvent)
     - Repository involved
     - Timestamp of the event
     - Brief description or summary of the event
   - **Purpose**: Provides insight into the user's recent activities and contributions.

2. **Repository Contributors Page**
   Description: Design a page that lists all contributors to a specific repository along with their contribution statistics.
   - **API Endpoint**: `GET /repos/{owner}/{repo}/contributors`
   - **Details to Display**
     - Contributor's username and avatar
     - Number of commits contributed
     - Link to the contributor's GitHub profile
   - **Purpose**: Highlights the collaborative nature of the repository and acknowledges contributors.

3. **Repository File Explorer**
   Description: Implement a file explorer interface that allows users to navigate through the repository's directory structure.
   - **API Endpoint**: `GET /repos/{owner}/{repo}/contents/{path}`
   - **Features**
     - Display folders and files hierarchically
     - Allow users to click on files to view their contents
     - Indicate file types with appropriate icons
   - **Purpose**: Provides an intuitive way to browse the repository's contents directly within the app.

4. **Repository Commit History Page**
   Description: Create a page that lists the commit history of a repository, allowing users to view recent changes.
   - **API Endpoint**: `GET /repos/{owner}/{repo}/commits`
   - **Details to Display**
     - Commit message
     - Author's name and avatar

    - Date of the commit
    - Link to the commit on GitHub
- **Purpose**: Enables users to track the evolution of the repository and understand recent developments.