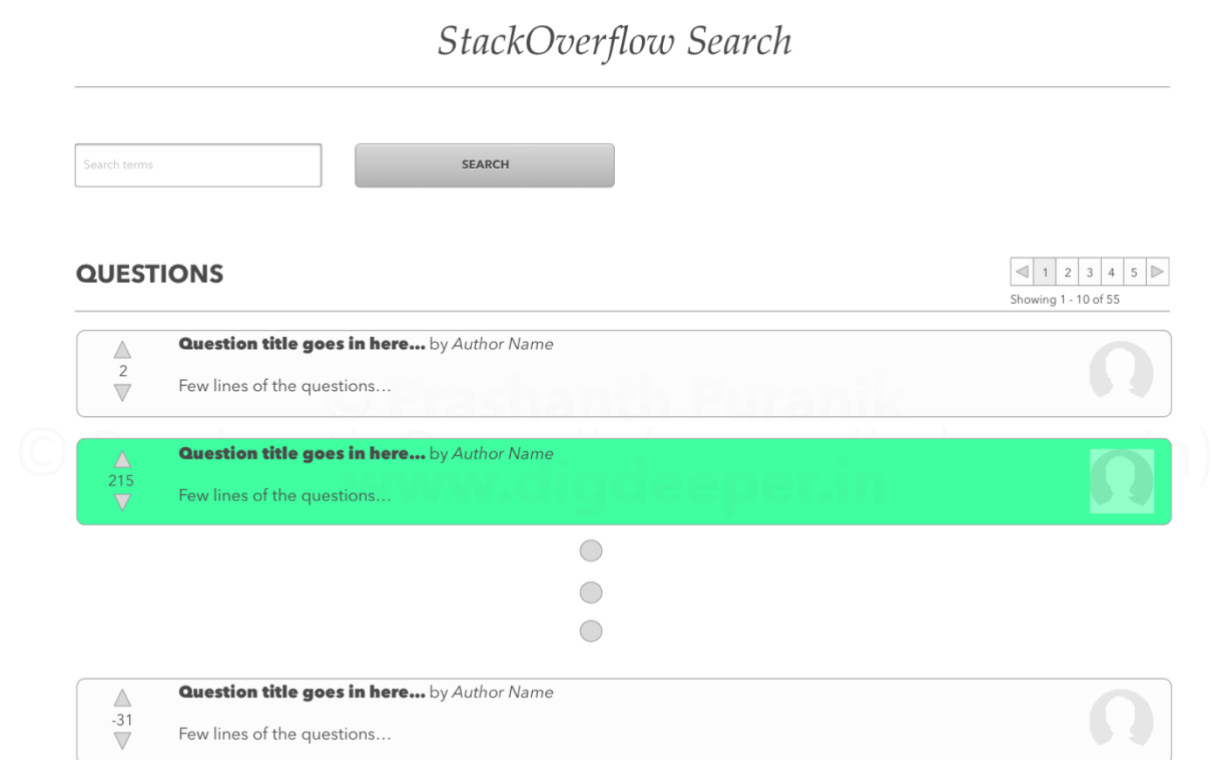


STACKOVERFLOW SEARCH APP

Use StackOverflow API (<https://api.stackexchange.com/>) to build a StackOverflow search app.

Mock-ups

Page 1: Search page (Home page)




[BACK TO SEARCH](#)

Question Title

▲

215

▼



Mark Smith

Details of the question go in here...

ANSWERS

◀

1

2

3

4

5

▶

Showing 1 - 10 of 55

▲

102

▼

Answer title goes in here... by Author Name

Answer text



▲

215

▼

Answer title goes in here... by Jane Doe

Answer text





▲

-7

▼

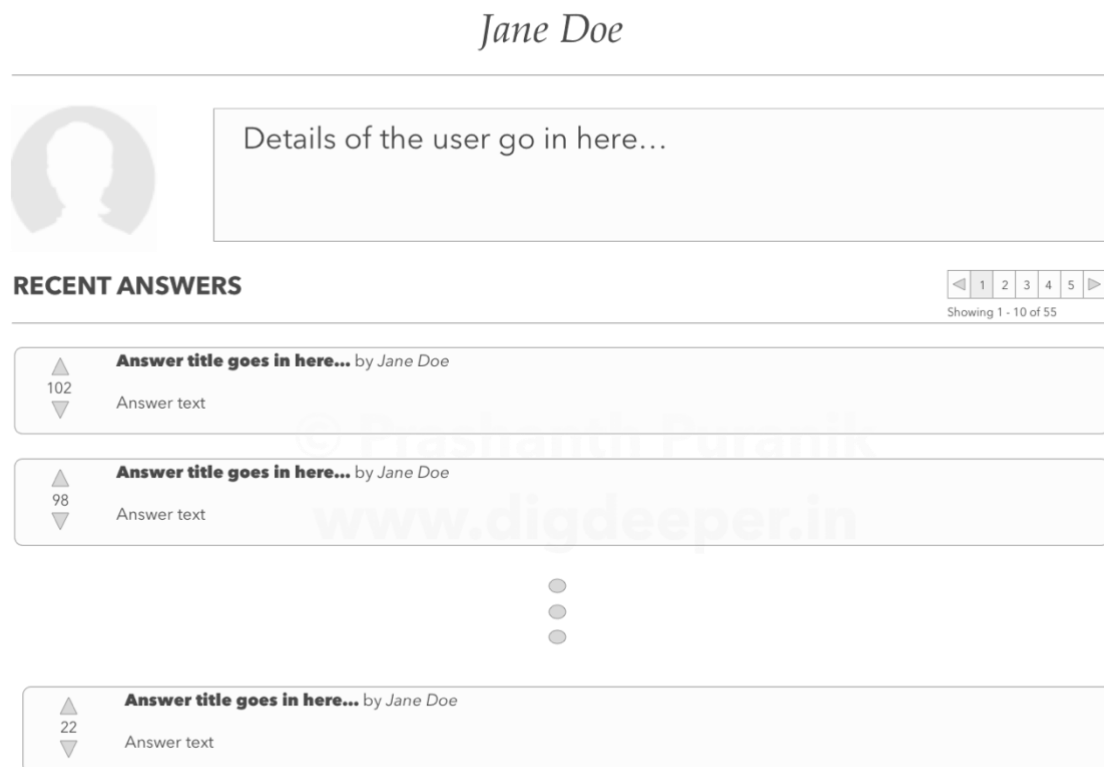
Answer title goes in here... by Author Name

Answer text



© Prashanth Puranik (www.digdeeper.in)

Page 3: User Details page



Functionality

1. On the home page, a list of questions sorted based on last_activity_date is shown by default. There is pagination with 10 results per page. Basic details of questions are shown.

The page has a search functionality. When the search button is clicked, it searches for questions where the question title has the supplied search terms and displays results in the table below. Clicking a question list item entry takes one to the question detail page. Show answered questions in green background and unanswered questions in lightgray background color.

2. The question details page has the questions details like title, the question text, upvotes, user who asked the question etc. Below that is a list of answers for the questions, along with author name, upvotes for the answer etc.

3. Throughout the app, user names that get displayed (eg. in question list, answers etc.) are linked to the user details page. This displays the details of the user, along with 10 recent answers.

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. Question Comments Section

Description: Enhance the question details page by displaying all comments associated with a question.

- **API Endpoint:** `GET /questions/{ids}/comments`
- **Details to Display**
 - Comment text
 - Author's name (linked to their profile)
 - Creation date
 - Score
- **Purpose:** Provides additional context and clarifications related to the question, enriching the user's understanding.

2. User's Asked Questions Page

Description: Create a dedicated page to list all questions asked by a specific user.

- **API Endpoint:** `GET /users/{ids}/questions`
- **Details to Display**
 - Question titles (linked to their details)
 - Creation dates
 - Scores
 - Answer counts
- **Purpose:** Allows users to explore the topics a particular user is interested in or has contributed to.

3. Tag Details Page

Description: Implement a page that provides information about a specific tag, including its description and related tags.

- **API Endpoints:**
 - `GET /tags/{tags}/info`
 - `GET /tags/{tags}/related`
- **Details to Display**
 - Tag description
 - Number of questions tagged
 - List of related tags
- **Purpose:** Helps users understand the context and usage of specific tags, promoting better tagging practices.

4. Related Questions Section

Description: On the question details page, display a list of questions that are related to the current question.

- **API Endpoint:** `GET /questions/{ids}/related`

- **Details to Display**
 - Related question titles (linked to their details)
 - Creation dates
 - Scores
- **Purpose:** Provides users with additional resources and discussions related to their query, enhancing the app's utility.

© Prashanth Puranik (www.digdeeper.in)