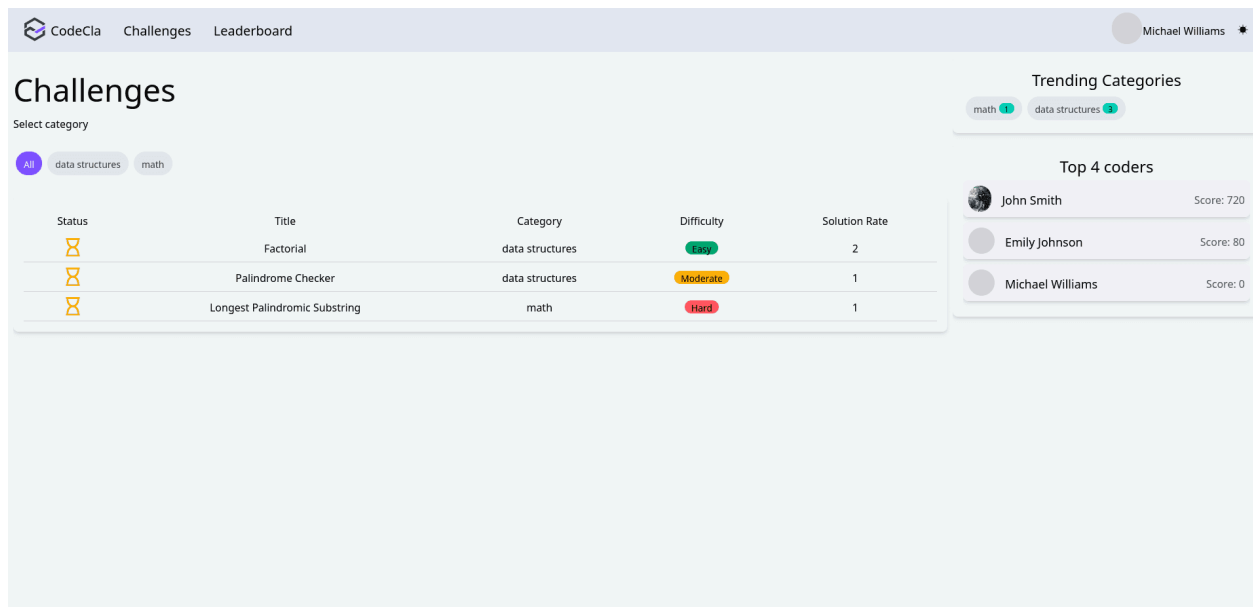


In this section of the assignment, you'll be tasked with implementing the home page. The home page should display general information about the platform, such as:

- List of challenges available.
- Categories of challenges.
- Top trending categories.
- Top 4 ranked coders.



## Tasks

Below are the tasks required to design the UI for the home page:

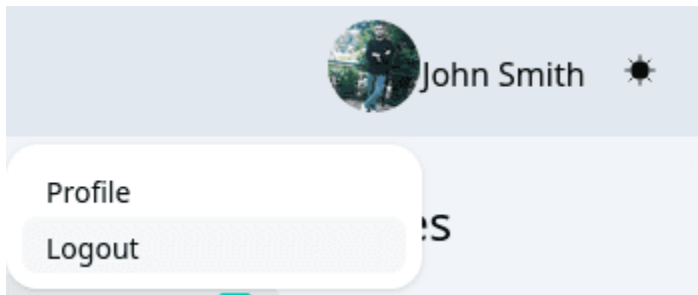
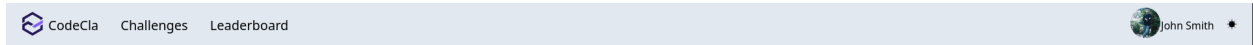
### 1. Implementing Components

#### 1.1. Navbar Component

To implement the Navbar component, follow these steps:

- Create a React component named Navbar.
- In the Navbar, include the following elements: Logo, Links for "Challenges" and "Leaderboard" on the left side, Profile avatar, user name, and theme toggle on the right side

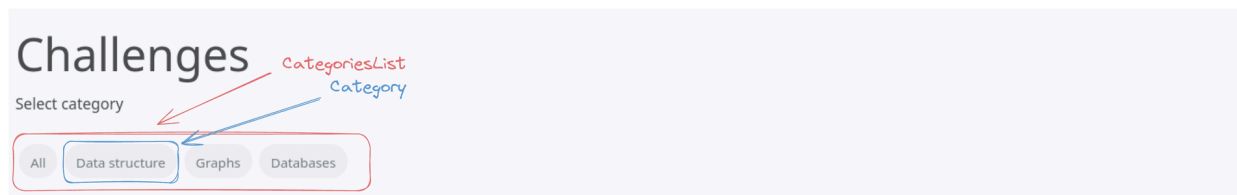
- Implement dropdown functionality for user profile options, which opens up when the user clicks on their profile avatar.



## 1.2. CategoriesList Component

To implement the CategoriesList component, follow these steps:

- Create a React component named CategoriesList, which hosts the list of challenges' categories in the platform.
- Use some categories dummy data, for example:  
**["Data structure", "Graphs", "Databases"]**
- Inside the CategoriesList component, map through the categories array and render individual Category components for each category






## 1.3. ChallengesList Component




- Create a react component named ChallengesList.
- Use dummy data to populate the challenges table. For example, you can use this data:

```
[
  {
    "id": 145,
    "title": "Two-sum",
    "description": "...", // Not used here
    "category": "Data structure",
    "Difficulty": "Easy",
    "status": "Completed",
    "solutionRate": "45%",
    "maintainer": "Goerge Harvy" // Not used here
  },
  {
    "id": 146,
    "title": "Fibonatci series",
    "description": "...", // Not used here
    "category": "Data structure",
    "Difficulty": "Moderate",
    "status": "Attempted",
    "solutionRate": "45%",
    "maintainer": "Goerge Harvy" // Not used here
  },
  {
    "id": 147,
    "title": "Skyline problem",
    "description": "...", // Not used here
    "category": "Data structure",
    "Difficulty": "Moderate",
    "status": "Pending",
    "solutionRate": "45%",
    "maintainer": "Goerge Harvy" // Not used here
  }
]
```

- Implement a scrollable container for the table, so that the table does not overflow from the page.

- For the icons you can use the **react-icons** library. We are using BsCheck2Circle/LuFileSpreadsheet/FaRegHourglass for a Completed/Attempted/Pending challenge
- Implement a tooltip feature so that when the user hovers over the status, they can see the description of the status, similar to the depiction in the second image.

Status	Title	Category	Difficulty	Solution Rate
	Two-sum	Data structure	Easy	45%
	Fibonacci series	Data structure	Moderate	45%
	Fibonacci series	Data structure	Hard	45%

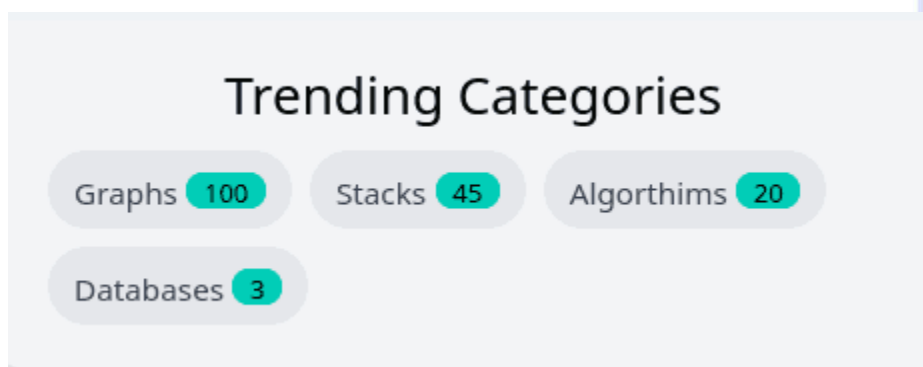
Status	Title	Category	Difficulty	Solution Rate
	Two-sum	Data structure	Easy	45%
	Fibonacci series	Data structure	Moderate	45%
	Skyline Problem	Data structure	Hard	45%

## 1.4. TrendingCategoriesBox Component

- Create a React component named TrendingCategoriesBox. This component should list the most submitted categories.
- You can use the following test data:

```
[
  {
    "category": "Graphs",
    "count": 100
  },
  {
    "category": "Stacks",
    "count": 45
  },
  {
    "category": "Algorithms",
    "count": 20
  },
  {
    "category": "Databases",
    "count": 3
  }
]
```

- As depicted in the following picture, the trending categories component should display each category along with the number of challenges available on the platform within that category.



### 1.5. TopKCodersList Component

- Create a React component named TopKCodersList that list coders cards.
- Use dummy data to populate the top coders list. Here's a sample data:

```
[
  {
    "id": 101,
    "first_name": "Alice",
    "last_name": "Johnson",
    "avatar_url": "https://i.pravatar.cc/150?img=1",
    "score": 350
  },
  {
    "id": 102,
    "first_name": "Bob",
    "last_name": "Smith",
    "avatar_url": "https://i.pravatar.cc/150?img=2",
    "score": 320
  },
  {
    "id": 103,
    "first_name": "Emily",
    "last_name": "Davis",
    "avatar_url": "https://i.pravatar.cc/150?img=5",
    "score": 290
  },
  {
    "id": 104,
    "first_name": "Michael",
    "last_name": "Brown",
    "avatar_url": "https://i.pravatar.cc/150?img=4",
    "score": 270
  }
]
```

- Implement CoderCard and make sure to show the avatar if it exists, an icon indicating its rank, its name and its score.

### Top 4 coders



Alice Johnson

Score: 350



Bob Smith

Score: 320



Emily Davis

Score: 290



Michael Brown

Score: 270

## 2. Update the Router

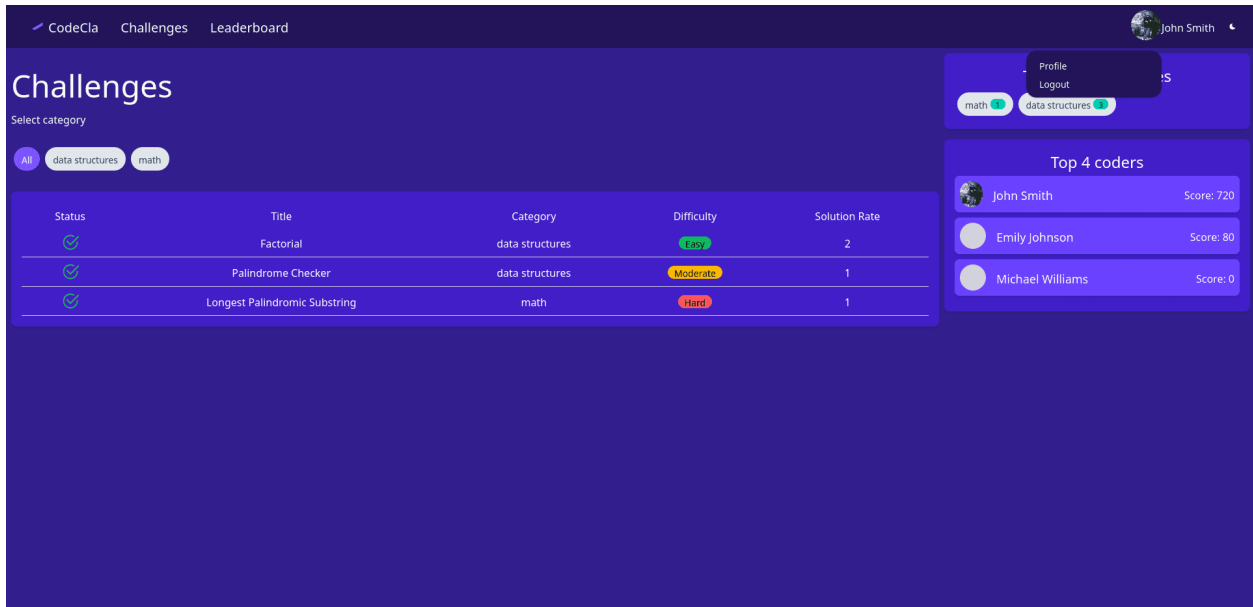
Make sure to layout these components in a page component named Home, which is the index page for the app. We can access it via two link paths: / and /challenges.

## 3. Theme Toggle

Next, you are asked to implement theme toggler feature.

There are two options to implement it, either using the context API or redux. In this assignment you are going to use redux to implement it.

- Add a slice for the theme in your redux store configuration.
- The initial state should be a value that indicates whether the application is in light or dark mode.
- Add the action reducer to toggle the theme.
- Add a toggle component to your Navbar, you can make use of react-theme-toggle library and attach a handler that dispatches the toggle action on click events.
- For the dark theme color, you can use the colors mentioned in the introduction.



The screenshot shows the 'Challenges' page of the Code Labs Academy. The page has a dark blue header with navigation links for 'CodeCla', 'Challenges', and 'Leaderboard'. A user profile for 'John Smith' is visible in the top right corner. Below the header, the 'Challenges' section is displayed with a 'Select category' dropdown and buttons for 'All', 'data structures', and 'math'. A table lists three challenges:

Status	Title	Category	Difficulty	Solution Rate
✓	Factorial	data structures	Easy	2
✓	Palindrome Checker	data structures	Moderate	1
✓	Longest Palindromic Substring	math	Hard	1

On the right side, there is a 'Top 4 coders' section showing a leaderboard:

Coder	Score
John Smith	720
Emily Johnson	80
Michael Williams	0

**Optional:** As an optional task, you can implement a feature to store the theme configuration in the user's browser. This way, when the user closes the tab or the browser, the theme will be initialized with their last configured theme. (Hint: Search for browsers support for storage).