

REPORT

“Reddit Feed Viewer- Fetching and Displaying posts from a chosen subreddit using the Reddit API”

Team Members:

Ansh Rathi - 3

Shreyas Desai - 7

Kartik Bankar - 19

Neeraj Gupta - 27

Himesh Pal - 30

Abstract

The Reddit Feed Viewer is a modern web application developed using React, Vite, and Tailwind CSS that enables users to browse and explore content from any subreddit on Reddit. The application provides a responsive, user-friendly interface that displays posts with their associated metadata while incorporating modern design principles and smooth interactions.

Introduction

Social media content aggregation has become increasingly important for users who want to stay updated with specific communities and topics. Reddit, as one of the largest social news and discussion platforms, contains vast amounts of user-generated content organized into subreddits. This project aims to create an intuitive interface for accessing and viewing Reddit content through Reddit's public API, making it easier for users to explore their favorite communities.

Background

Reddit provides a JSON API that allows developers to access public content without requiring authentication for basic read operations. This creates an opportunity to build applications that can fetch and display Reddit content in custom interfaces. The project leverages modern web technologies:

- React: A JavaScript library for building user interfaces
- Vite: A modern frontend build tool that offers superior development experience
- Tailwind CSS: A utility-first CSS framework for rapid UI development
- Lucide React: A comprehensive icon library for visual elements

System Description

Architecture Overview:

The application follows a component-based architecture typical of React applications, with the following key elements:

1. Data Flow

- User input handling for subreddit selection
- API integration for fetching Reddit data
- State management using React hooks
- Data transformation and formatting

2. User Interface

- Search interface for subreddit selection

- Post display system
- Loading states and error handling
- Responsive design

elements Core Modules:

1. Search Module

- Input handling and validation
- Form submission management
- Real-time input state updates
- Error boundary for invalid inputs

2. Data Fetching Module

- API endpoint construction
- Asynchronous data fetching
- Response parsing and error handling
- Data transformation for display

3. Post Display Module

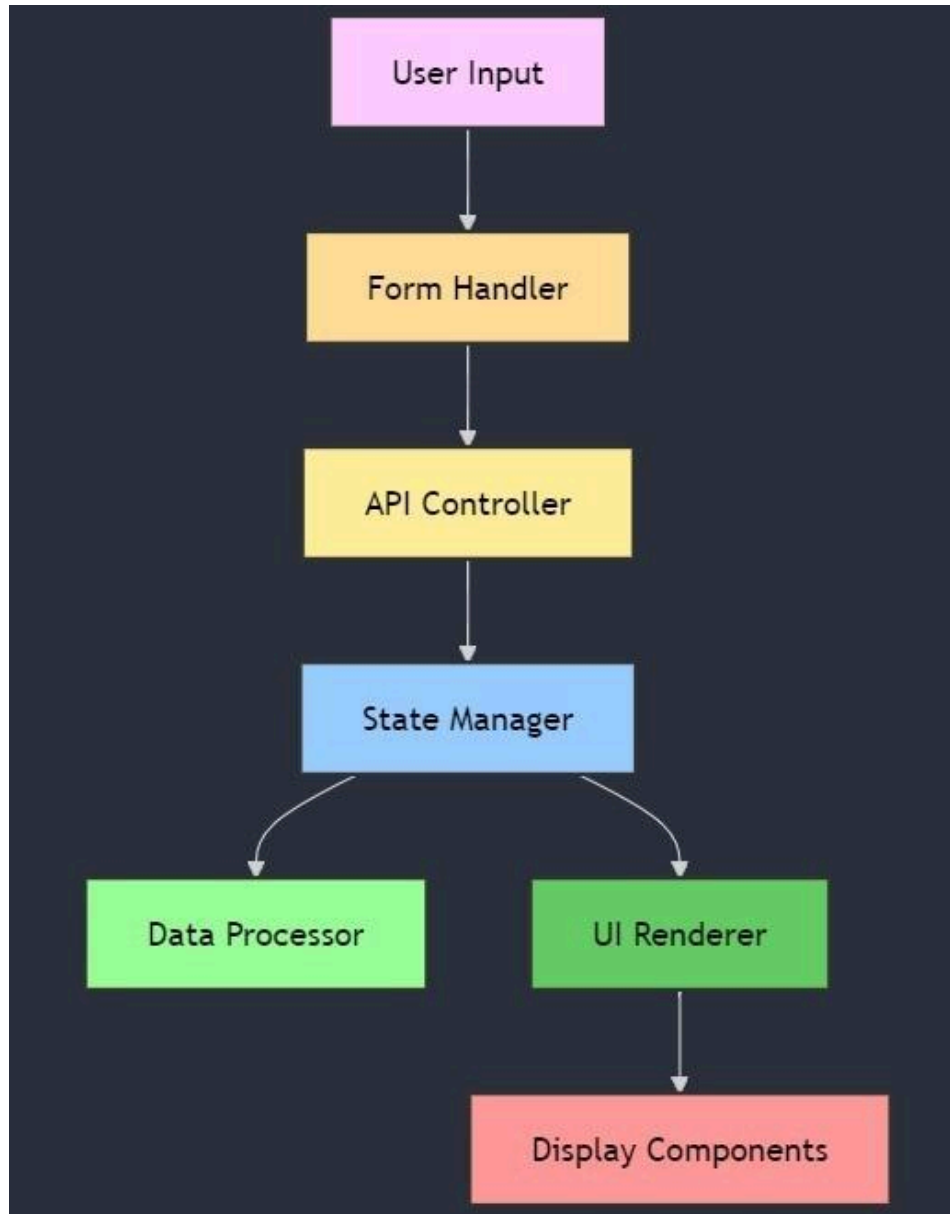
- Post card component
- Metadata formatting (timestamps, numbers)
- Interactive elements (links, hover states)
- Responsive layout management

4. UI/UX Module

- Gradient backgrounds and effects
- Animation system
- Icon integration
- Typography

hierarchy Block

Diagram:



Key Features:

1. Search Functionality
 - Real-time input validation
 - Subreddit name processing
 - Search history management
 - Error feedback system
2. Post Management
 - Title display with links
 - Vote count visualization

- Comment count tracking
- Timestamp formatting
- Author information display

3. Visual Design

- Gradient background system
- Card-based post layout
- Shadow and elevation effects
- Responsive grid system
- Interactive hover states
- Loading animations

4. Error Handling

- Invalid subreddit detection
- Network error management
- User feedback system
- Graceful fallback states

The system is designed to be scalable and maintainable, with clear separation of concerns between data management, user interface, and business logic. Each module is built to be independent yet interconnected, allowing for easy updates and feature additions in the future.

Future enhancements could include:

- Post content preview capabilities
- Advanced sorting and filtering options
- User authentication for additional features
- Offline support through caching
- Advanced analytics and tracking
- Multi-subreddit aggregation
- Custom theming options
- Bookmark and save functionality

The application successfully demonstrates the integration of modern web technologies to create a performant and user-friendly interface for browsing Reddit content, with room for expansion and improvement based on user feedback and requirements.

live link -> <https://redditfeedviewer.netlify.app/>

Accessible subreddits -> <https://www.reddit.com/subreddits/>

Code:

```
const RedditFeedViewer = () => {
  const [subreddit, setSubreddit] = useState("reactjs");
  const [posts, setPosts] = useState([]);
  const [loading, setLoading] = useState(false);
  const [error, setError] = useState("");
  const [inputValue, setInputValue] = useState("reactjs");

  useEffect(() => {
    fetchPosts();
  }, [subreddit]);

  const fetchPosts = async () => {
    setLoading(true);
    setError("");
    try {
      const response = await fetch(
        https://www.reddit.com/r/${subreddit}/hot.json
      );
      if (!response.ok) throw new Error("Subreddit not found or private");
      const data = await response.json();
      setPosts(data.data.children);
    } catch (err) {
      setError("Error fetching posts. Please check the subreddit name.");
      setPosts([]);
    }
    setLoading(false);
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    if (inputValue.trim()) {
      setSubreddit(inputValue.trim());
    }
  };

  const formatTime = (timestamp) => {
    const hours = Math.floor((Date.now() / 1000 - timestamp) / 3600);
    if (hours < 24) return `${hours}h ago;

```

```

const days = Math.floor(hours / 24);
return ${days}d ago;
};

return (
  <div className="min-h-screen bg-gradient-to-br from-indigo-50 via-purple-50 to-pink-50">
    <div className="max-w-4xl mx-auto p-6">
      { /* Header */ }
      <div className="flex items-center gap-2 mb-8">
        <Sparkles className="text-purple-500" size={24} />
        <h1 className="text-2xl font-bold bg-gradient-to-r from-purple-600 to-pink-600
text-transparent bg-clip-text">
          Reddit Explorer
        </h1>
      </div>

      { /* Search Form */ }
      <div className="mb-8">
        <form onSubmit={handleSubmit} className="flex gap-3">
          <div className="relative flex-1">
            <Search
              className="absolute left-4 top-1/2 transform -translate-y-1/2 text-gray-400"
              size={20}
            />
            <input
              type="text"
              value={inputValue}
              onChange={(e) => setInputValue(e.target.value)}
              placeholder="Enter subreddit name..."
              className="w-full pl-12 pr-4 py-3 border-0 rounded-xl bg-white shadow-lg
focus:outline-none focus:ring-2 focus:ring-purple-500 transition-shadow"
            />
          </div>
          <button
            type="submit"
            className="px-6 py-3 bg-gradient-to-r from-purple-600 to-pink-600 text-white
rounded-xl hover:opacity-90 transition-opacity shadow-lg"
          >
            Explore
          </button>

```

</form>

</div>

{/* Current Subreddit Display */}

<div className="mb-6">

Currently browsing

<h2 className="text-xl font-semibold text-gray-800">r/{subreddit}</h2>

</div>

{loading && (

<div className="flex justify-center items-center py-12">

<div className="animate-spin rounded-full h-10 w-10 border-4 border-purple-500 border-t-transparent"></div>

</div>

))

{error && (

<div className="bg-red-50 border-l-4 border-red-500 p-4 rounded-lg mb-6">

<div className="flex">

<div className="flex-shrink-0">

<svg

className="h-5 w-5 text-red-400"

viewBox="0 0 20 20"

fill="currentColor"

>

<path

fillRule="evenodd"

d="M10 18a8 8 0 100-16 8 8 0 000 16zM8.707 7.293a1 1 0 00-1.414 1.414L8.586

101-1.293 1.293a1 1 0 101.414 1.414L10 11.41411.293 1.293a1 1 0 001.414-1.414L11.414

1011.293-1.293a1 1 0 00-1.414-1.414L10 8.586 8.707 7.293z"

clipRule="evenodd"

/>

</svg>

</div>

<div className="ml-3">

<p className="text-sm text-red-700">{error}</p>

</div>

</div>

</div>

))

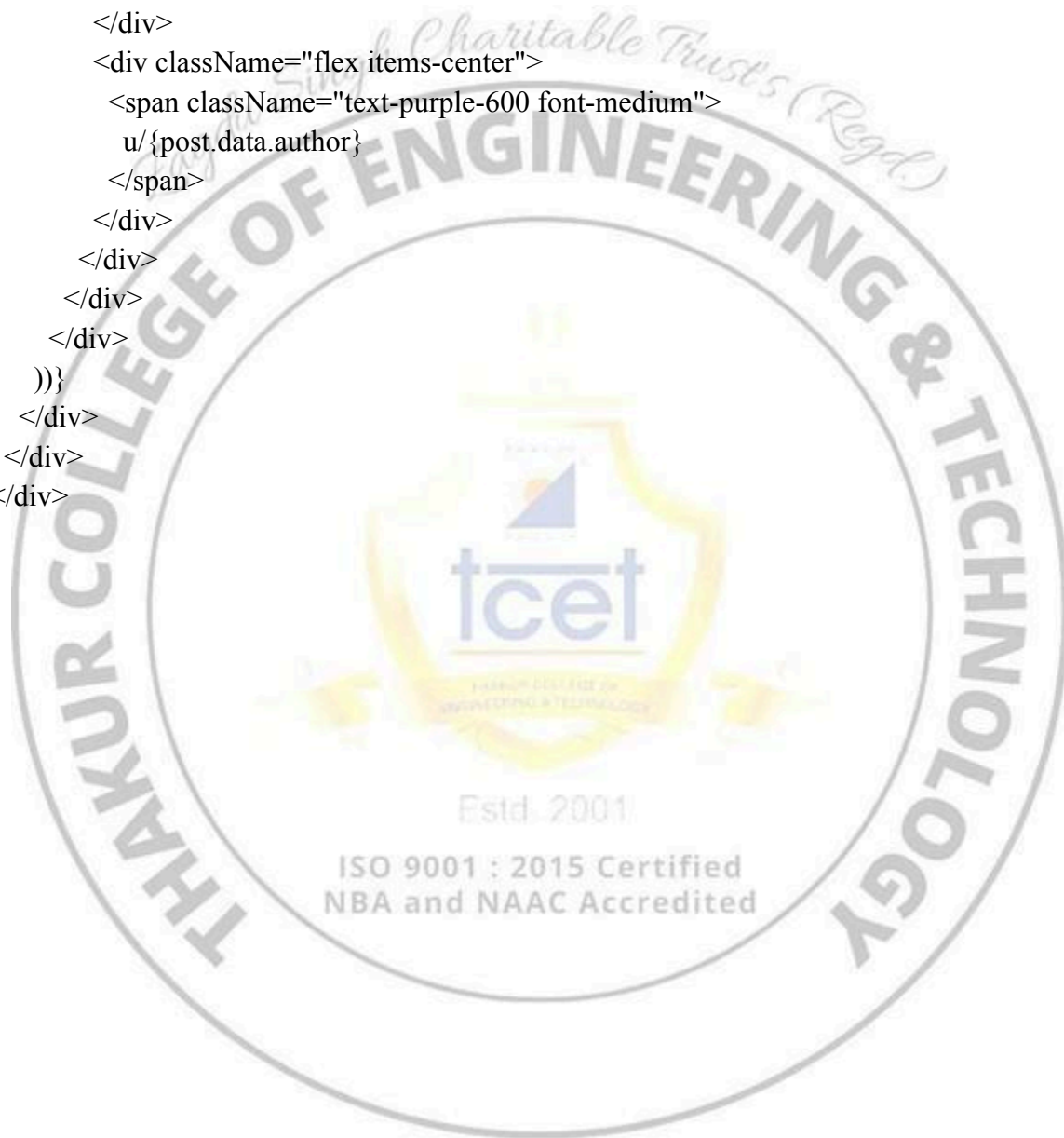

```
<div className="space-y-6">
  {posts.map((post) => (
    <div
      key={post.data.id}
      className="bg-white rounded-xl shadow-lg hover:shadow-xl transition-all
duration-300 overflow-hidden group"
    >
      <div className="p-6">
        <div className="flex items-start justify-between">
          <h3 className="text-lg font-semibold mb-3
group-hover:text-purple-600 transition-colors">
            <a
              href={https://reddit.com${post.data.permalink}}
              target="_blank"
              rel="noopener noreferrer"
            >
              {post.data.title}
            </a>
          </h3>
          <a
            href={https://reddit.com${post.data.permalink}}
            target="_blank"
            rel="noopener noreferrer"
            className="text-gray-400 hover:text-purple-500 transition-colors"
          >
            <ExternalLink size={16} />
          </a>
        </div>

        <div className="flex items-center gap-6 text-sm text-gray-500">
          <div className="flex items-center gap-1.5">
            <ArrowUpCircle size={18} className="text-orange-500" />
            <span className="font-medium">
              {post.data.ups.toLocaleString()}
            </span>
          </div>
          <div className="flex items-center gap-1.5">
            <MessageCircle size={18} className="text-blue-500" />
            <span className="font-medium">
```

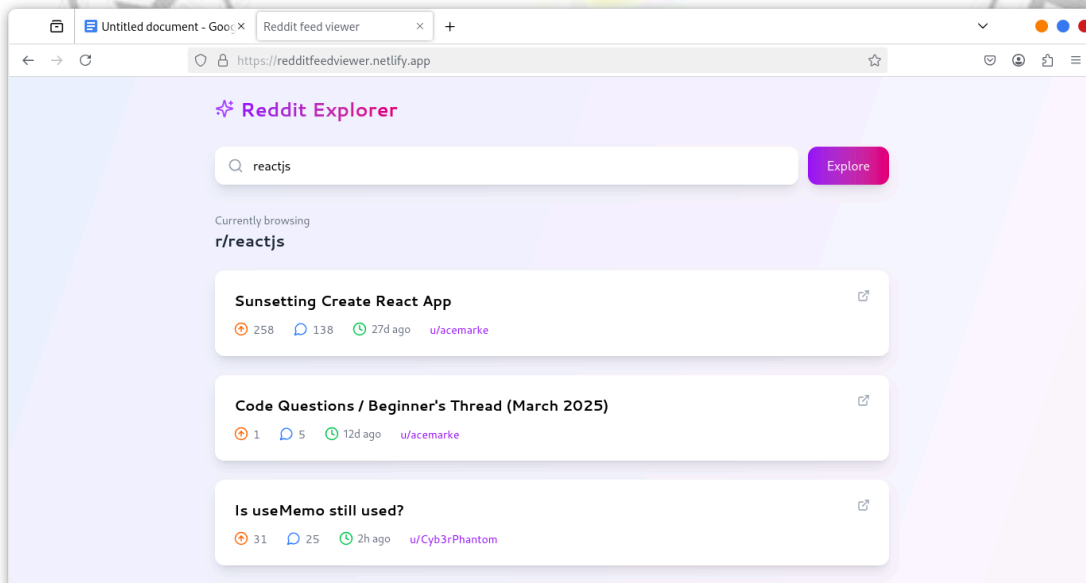
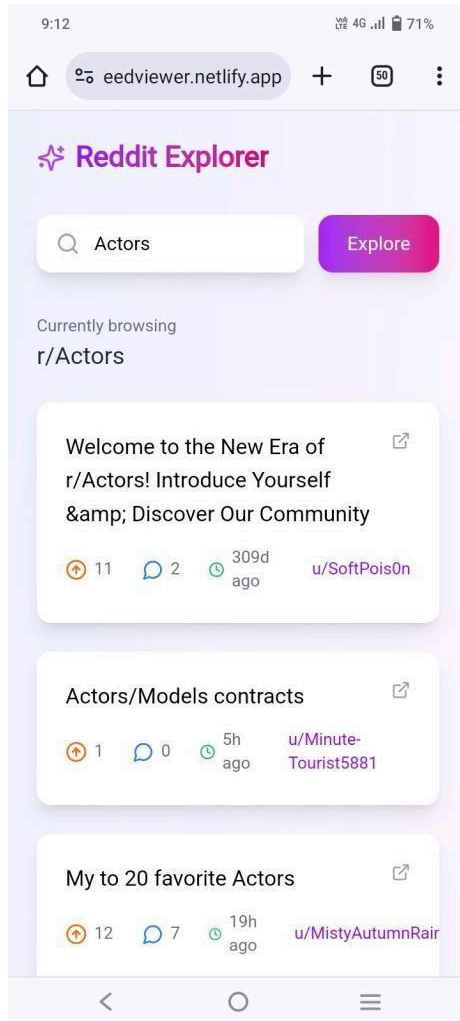
```

        {post.data.num_comments.toLocalizedString()}
    </span>
</div>
<div className="flex items-center gap-1.5">
    <Clock size={18} className="text-green-500" />
    <span>{formatTime(post.data.created_utc)}</span>
</div>
<div className="flex items-center">
    <span className="text-purple-600 font-medium">
        u/{post.data.author}
    </span>
</div>
</div>
</div>
</div>
</div>
    )}
</div>
</div>
</div>
);
};

```



Result and Discussion



The Reddit Feed Viewer successfully integrates Reddit's public API with a modern frontend stack, providing a seamless and interactive browsing experience for users. Key findings from the development and usage of the application are discussed below:

Performance and Responsiveness

- The use of **Vite** significantly improved development speed and build performance compared to traditional React setups.
- **Tailwind CSS** allowed for rapid UI development and ensured a consistent design system with minimal CSS bloat.
- The application maintains smooth navigation and fast load times due to efficient API calls and state management using **React hooks**.

User Experience and Accessibility

- The **search module** allows users to quickly find subreddits with real-time validation and feedback.
- The **post display module** provides a clean and structured presentation of Reddit posts, incorporating metadata like upvotes, comments, and timestamps.
- **Error handling** ensures users receive meaningful messages when encountering invalid subreddit names or network issues.

Scalability and Maintainability

- The **modular architecture** ensures that different parts of the application can be updated or replaced independently.
- Future features, such as **authentication**, **caching**, and **post previews**, can be easily integrated without disrupting existing functionality.

Conclusion and Future Scope

Conclusion

The Reddit Feed Viewer successfully demonstrates the power of **React**, **Vite**, and **Tailwind CSS** in building a lightweight yet functional web application for browsing Reddit content. By leveraging Reddit's JSON API, users can effortlessly explore various subreddits in a structured and visually appealing manner.

The application effectively handles:

- **Subreddit selection and validation**
- **Asynchronous data fetching and transformation**
- **User-friendly UI with responsive design**
- **Error handling and feedback mechanisms**

Future Scope

Several enhancements can be implemented to improve the application's capabilities:

1. **Post Content Preview** – Display a snippet of post content for better readability.
2. **Sorting and Filtering** – Allow users to filter posts by trending, top, new, or controversial categories.
3. **User Authentication** – Enable users to log in for personalized feeds and interactions.
4. **Offline Support** – Implement caching mechanisms to allow content browsing without an active internet connection.
5. **Multi-Subreddit Aggregation** – Allow users to view posts from multiple subreddits at once.
6. **Theming and Customization** – Introduce dark mode and customizable UI settings.
7. **Bookmark and Save Functionality** – Let users save favorite posts for later viewing.
8. **Analytics and Tracking** – Provide insights into user interaction patterns.

By incorporating these features, the Reddit Feed Viewer can evolve into a more comprehensive Reddit browsing tool.

References

1. **Reddit API Documentation** – <https://www.reddit.com/dev/api>
2. **React Documentation** – <https://react.dev/>
3. **Vite Documentation** – <https://vitejs.dev/>
4. **Tailwind CSS Documentation** – <https://tailwindcss.com/docs>
5. **Lucide React Icons** – <https://lucide.dev/docs/libraries/react>
6. **Netlify Deployment Guide** – <https://docs.netlify.com>