CodeRec

## MINI PROJECT–II SYNOPSIS



Department of Computer Science & Application

## Institute of Engineering & Technology

SUBMITTED TO: - SUBMITTED BY: -

Mr. Mayank Saxena Pavan Gupta (201500467)

(Technical Trainer) Kshitiz Kumar Singh (201500357)

Mohit Varshney (201500409)

Utkarsh Srivastav (201500762)

# Acknowledgement

It gives us a great sense of pleasure to present the synopsis of the B.Tech mini project undertaken during B.Tech III Year. This project is going to be an acknowledgement to the inspiration, drive and technical assistance will be contributed to it by many individuals. We owe special debt of gratitude to Mr. Mayank Saxena, for providing us with an encouraging platform to develop this project, which thus helped us in shaping our abilities towards a constructive goal and for his constant support and guidance to our work.

His sincerity, thoroughness and perseverance has been a constant source of inspiration for us. We believe that he will shower us with all her extensively experienced ideas and insightful comments at different stages of the project & also taught us about the latest industry-oriented technologies. We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind guidance and co-operation.

Pavan Gupta (201500467)

Kshitiz Kumar Singh (201500357)

Mohit Varshney (201500409)

Utkarsh Srivastav (201500762)

## ABSTRACT

CodeRec is a web application that enables users to compare their coding skills and achievements with other programmers from GitHub, LeetCode, and CodeChef. By entering the usernames or profile URLs of the users they want to compare, CodeRec fetches the user data from each platform and displays it in charts and graphs for easy comparison. CodeRec uses React for the frontend, Node.js and Express for the backend, GitHub API for data retrieval, and Chart.js for data visualization. The motivation for this project was to provide a tool for programmers to benchmark their coding abilities and learn from others. The main challenge was to handle the different APIs and data formats of each platform. The future plans for this project include adding more features like filtering, sorting, and sharing the comparison results.

# Contents

Abstract Declaration Acknowledgement

1. Introduction
   1. Objective
   2. Motivation
   3. Problem Statement
2. Software Requirement
   1. Hardware Requirements
   2. Software Requirements
3. Project Description
4. Working
5. Implementation
6. References

# INTRODUCTION

Coding is a skill that requires constant practice and improvement. Programmers often use online platforms like GitHub, LeetCode, and CodeChef to showcase their coding projects, solve coding challenges, and participate in coding contests. These platforms provide various metrics and statistics to measure the coding skills and achievements of the users, such as number of repositories, commits, stars, followers, problems solved, rating, rank, etc.

However, these platforms have different APIs and data formats, which makes it difficult to compare the user data across them. Moreover, there is no existing tool that allows users to compare their coding skills and achievements with other programmers from multiple platforms in a convenient and comprehensive way.

CodeRec is a web application that aims to fill this gap by enabling users to compare their coding skills and achievements with other programmers from GitHub, LeetCode, and CodeChef. By entering the usernames or profile URLs of the users they want to compare, CodeRec fetches the user data from each platform and displays it in charts and graphs for easy comparison. CodeRec uses React for the frontend, Node.js and Express for the backend, GitHub API for data retrieval, and Chart.js for data visualization.

The objectives of this project are:

* To provide a user-friendly interface for users to enter the usernames or profile URLs of the users they want to compare
* To fetch the user data from GitHub, LeetCode, and CodeChef using their respective APIs
* To display the user data in charts and graphs for easy comparison using Chart.js
* To add more features like filtering, sorting, and sharing the comparison results

The scope of this project is limited to comparing the user data from GitHub, LeetCode, and CodeChef. Other online platforms or metrics are not considered in this project

## SOFTWARE AND HARDWARE REQUIREMENTS

* HTML5
* CSS3
* JavaScript
* React
* Node.js
* Express.js
* GitHub API

## PROJECT DESCRIPTION

CodeRec is a web application that enables users to compare their coding skills and achievements with other programmers from GitHub, LeetCode, and CodeChef. It is designed to help programmers benchmark their coding abilities and learn from others.

The tech stacks used for this project are:

* Frontend: React, HTML, CSS, JavaScript
* Backend: Node.js, Express
* GitHub API: to fetch user data
* Chart.js: to create charts and graphs for the comparison data

The design and architecture of this project are as follows:

* The frontend consists of a single-page application that uses React to create the user interface. The user interface consists of three main components: a form component, a comparison component, and a chart component.
* The form component allows the user to enter the usernames or profile URLs of the users they want to compare. The user can enter up to four users from each platform. The form component validates the input and sends it to the backend using an HTTP POST request.
* The backend consists of a server that uses Node.js and Express to handle the requests from the frontend. The server uses GitHub API to fetch the user data from GitHub, LeetCode, and CodeChef.

The server parses and processes the user data and sends it back to the frontend using an HTTP response.

* The comparison component receives the user data from the backend and displays it in a table format for easy comparison. The comparison component shows the metrics and statistics of each user from each platform, such as number of repositories, commits, stars, followers, problems solved, rating, rank, etc.
* The chart component receives the user data from the backend and displays it in charts and graphs for easy visualization. The chart component uses Chart.js to create different types of charts and graphs for the comparison data, such as bar charts, pie charts, line charts, etc.

The challenges and solutions of this project are as follows:

* The main challenge was to handle the different APIs and data formats of each platform. Each platform has its own API documentation, authentication method, rate limit, data structure, etc. To overcome this challenge, I had to study the API documentation of each platform carefully and use appropriate libraries and modules to fetch and parse the user data.
* Another challenge was to display the user data in charts and graphs for easy visualization. Each platform has different metrics and statistics that may not be directly comparable or compatible with each other. To overcome this challenge, I had to use Chart.js to create different types of charts and graphs for the comparison data and customize them according to the data type and format.

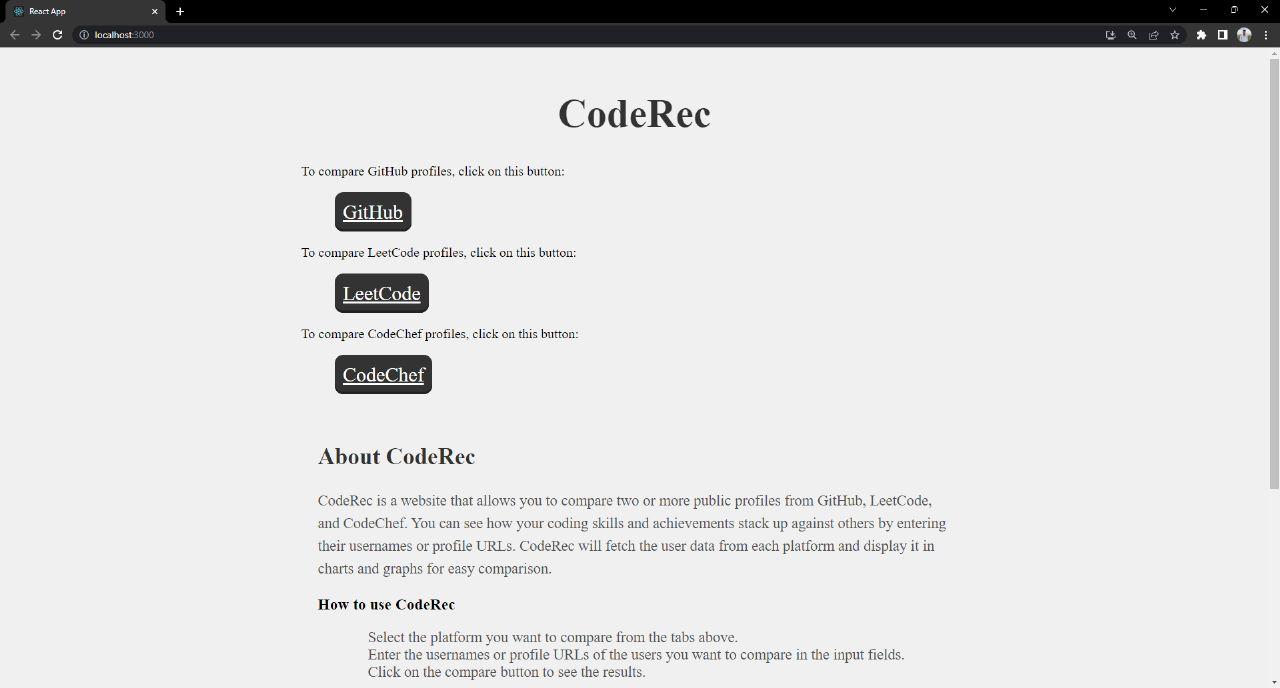
## Working

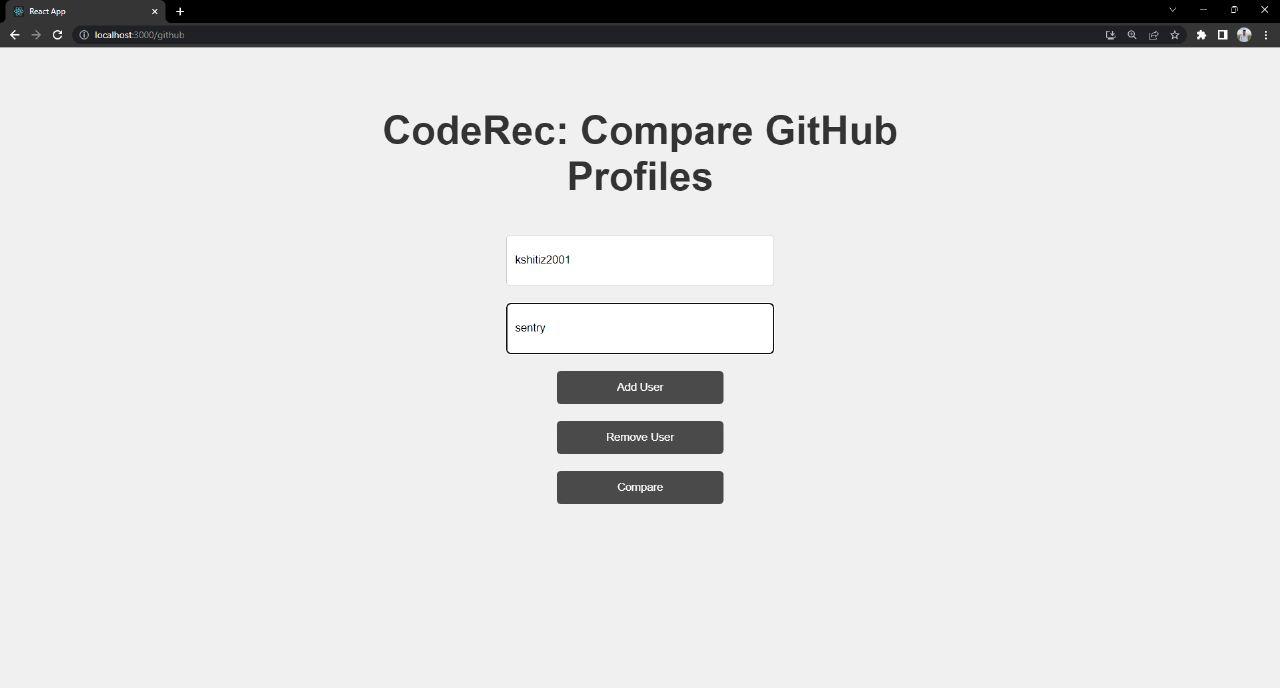
## To use CodeRec, the user needs to follow these steps:

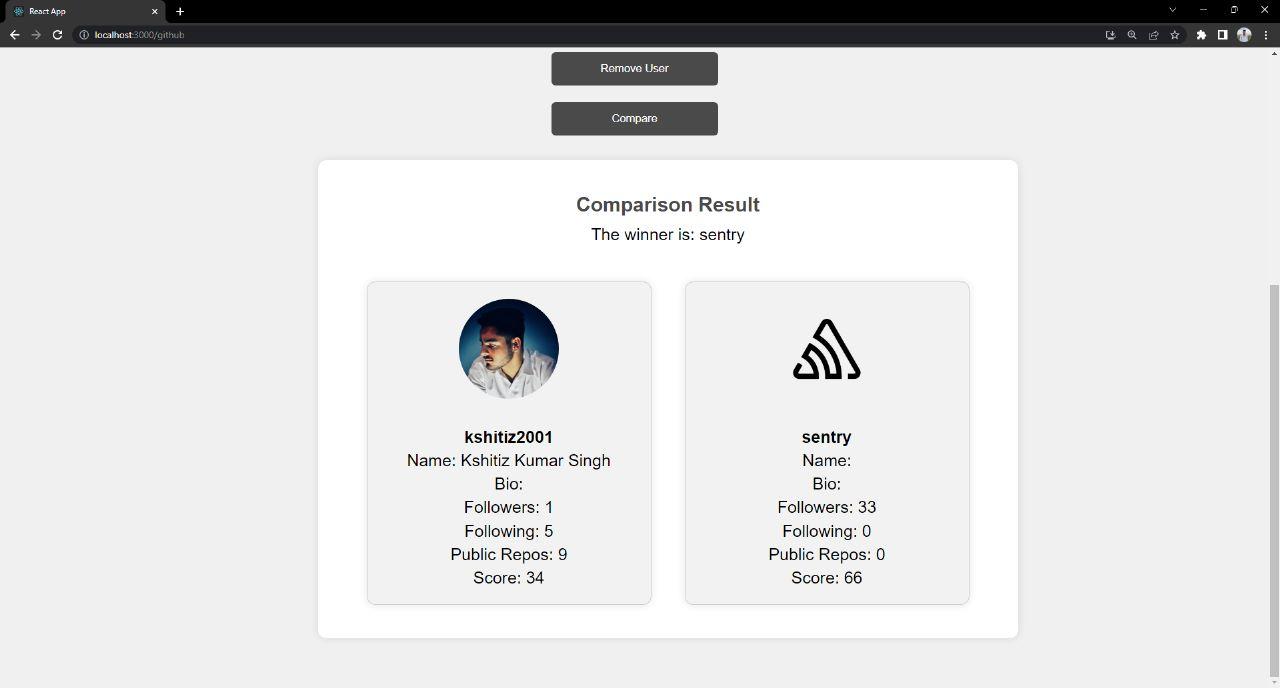
* Open the web application in a browser
* Enter the usernames or profile URLs of the users they want to compare in the form component. The user can enter up to four users from each platform.
* Click on the compare button to submit the form
* Wait for the user data to be fetched and displayed in the comparison component and the chart component
* View and analyze the comparison data in the table format and the charts and graphs
* Use the filter and sort options to refine the comparison data
* Share the comparison results with others using the share button

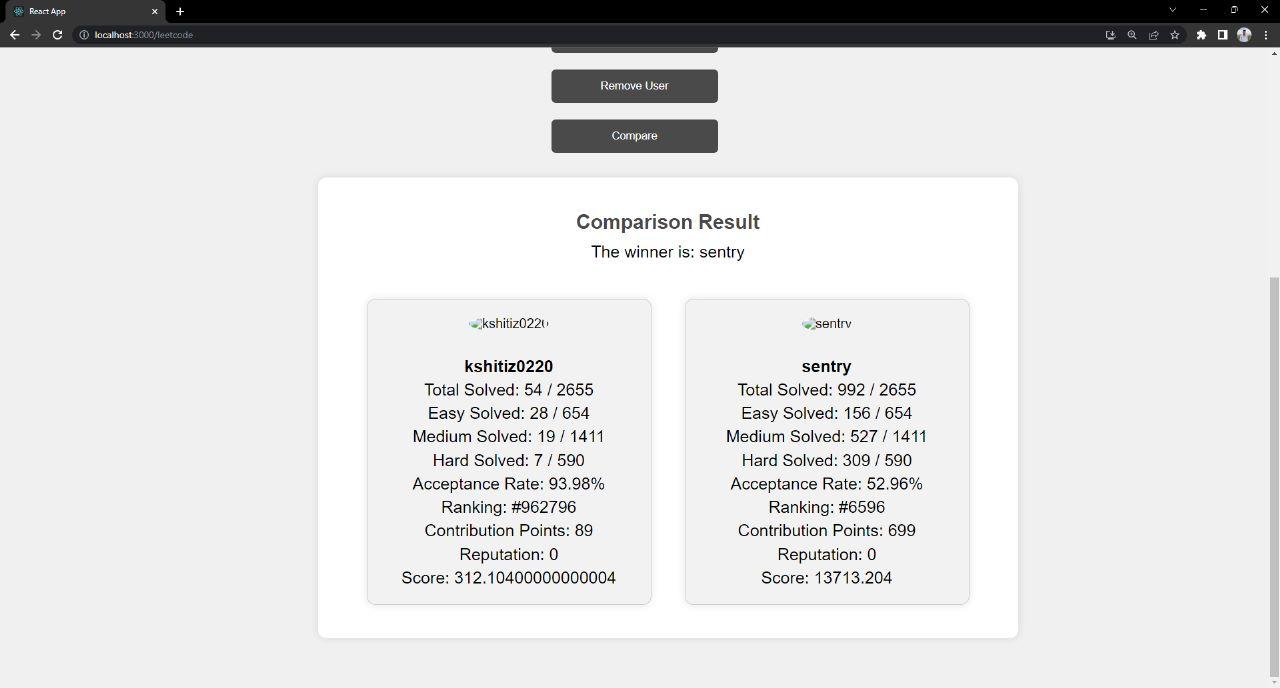
The expected output of CodeRec is a web page that shows the comparison data of the users from GitHub, LeetCode, and CodeChef in a table format and in charts and graphs. The user can view and analyze the comparison data and learn from others.

The screenshots or videos of CodeRec are as follows:







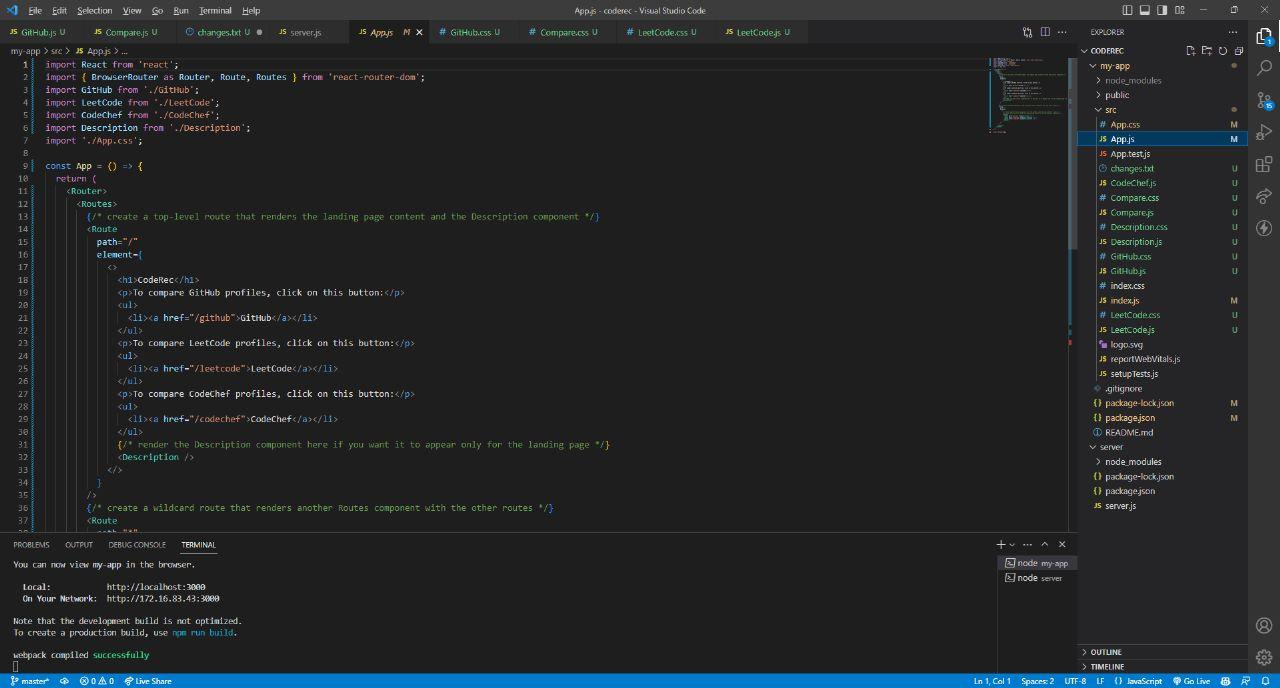


## IMPLEMENTATION

To implement CodeRec, I used the following tools and technologies:

* Frontend: React, HTML, CSS, JavaScript
* Backend: Node.js, Express
* GitHub API: to fetch user data
* Chart.js: to create charts and graphs for the comparison data

The code snippets of CodeRec are as follows:



The libraries and modules of CodeRec are as follows:

* React: a JavaScript library for building user interfaces
* HTML: a markup language for creating web pages
* CSS: a style sheet language for designing web pages
* JavaScript: a scripting language for adding interactivity to web pages
* Node.js: a JavaScript runtime environment for executing JavaScript code outside the browser
* Express: a web framework for Node.js
* GitHub API: an application programming interface for accessing GitHub data
* Chart.js: a JavaScript library for creating charts and graphs

The algorithms and data structures of CodeRec are as follows:

* To fetch the user data from GitHub, LeetCode, and CodeChef, I used the HTTP request and response methods of Node.js and Express. I used the axios module to make HTTP requests to the GitHub API and the cheerio module to scrape the HTML pages of LeetCode and CodeChef. I used the JSON.parse method to parse the JSON data from GitHub and the cheerio.load method to parse the HTML data from LeetCode and CodeChef. I used the try-catch blocks to handle any errors or exceptions that may occur during the data fetching process.
* To display the user data in charts and graphs, I used the Chart.js library. I created different types of charts and graphs for the comparison data, such as bar charts, pie charts, line charts, etc. I used the Chart constructor to create a new chart object and passed in the canvas element, the chart type, and the chart data as parameters. I used the options property to customize the appearance and behavior of the charts and graphs. I used the update method to update the charts and graphs when the user data changes.

The testing and debugging methods of CodeRec are as follows:

* To test the functionality and usability of CodeRec, I used various test cases and scenarios. I entered different usernames or profile URLs of different users from different platforms and checked if the user data was fetched and displayed correctly. I also checked if the filter and sort options worked as expected. I also checked if the share button generated a valid link to share the comparison results with others.
* To debug any errors or bugs in CodeRec, I used various tools and techniques. I used the console.log method to print out any messages or values to the console. I used the debugger statement to pause the execution of the code at a specific point. I used the Chrome DevTools to inspect and modify the elements, styles, sources, network.

## REFERENCES;

**Books:**

* Designing and Web Standards

-Jeffrey Jeldman with Ethan Marcotte

* React in Action

-Mark Teileins Thomas (2018)

* Full stack React: The Complete Guide to ReactJS and Friends

-Nate Murray (2017)

## Websites:

* <https://developer.mozilla.org/en-US/>
* [www.google.com](http://www.google.com/)
* [www.w3schools.com](http://www.w3schools.com/)
* https://reactjs.org
* [www.projectdeveloper.com](http://www.projectdeveloper.com/)
* [www.udemy.com/](http://www.udemy.com/)

## Faculty Guidelines:

Mr. Mayank Saxena (Technical Trainer in GLA University)

## link:

<https://github.com/pavangupta352/CodeRec>

<https://drive.google.com/drive/folders/1vqa8UWQf1HYoHA5yo5AoiTuTWg0iz6yp?usp=share_link>