# Scoop Labs Project Lab

### Full Stack Development Project Brief

**Project Title:** GitHub Profile Analyzer
**Difficulty Level:** Intermediate
**Estimated Duration:** 3–4 Weeks
**Suggested Stack:** React + GitHub REST API

---

## Project Overview

In this project, you will build a **GitHub Profile Analyzer web application** that fetches and displays data from the public GitHub API. The application will allow users to enter a GitHub username and view important profile information such as repositories, contribution statistics, programming languages used, and other developer insights.

The goal of this project is to help students understand how to **integrate external APIs, handle asynchronous data requests, and visualize data using modern web technologies**.

This project uses the **public GitHub API**, which is free to use for development purposes, allowing students to build and test the application without requiring paid services.

## Why This Project Matters

Modern web applications often rely on **external APIs to retrieve real-time data**. Understanding how to fetch, process, and display this data is an important skill for developers.

By completing this project, students will gain experience working with **real-world developer data from GitHub**, building interactive dashboards, and handling dynamic data in a frontend application.

This project also creates a **unique and practical portfolio project** that demonstrates API integration skills, which are highly valued by recruiters and technical interviewers.

# Learning Outcomes

By completing this project, students will learn and practice:

- Integrating external APIs in a web application
- Fetching and handling asynchronous data using JavaScript
- Working with **REST APIs**
- Managing application state in a React application
- Visualizing data using charts or graphical components
- Handling loading states and error handling in applications
- Building interactive dashboards using modern frontend tools

# Project Requirements

Your application should include the following functionality:

### GitHub User Search

- Input field where users can enter a GitHub username
- Fetch user data using the **GitHub REST API**

### Profile Information Display

- Display user's profile picture
- Show username and bio
- Display number of followers and following

### Repository Information

- Display a list of public repositories
- Show repository details such as stars, forks, and descriptions

### Programming Language Analysis

- Analyze repositories to identify most frequently used programming languages
- Display language statistics visually using charts or graphs

### Contribution and Activity Insights

• Show contribution activity or repository statistics where possible

### Responsive Interface

• The application should work properly on **desktop and mobile devices**

# Technical Guidelines

Students are encouraged to follow the technical guidelines below while implementing the project.

### Frontend

• Use **React** to build the user interface
• Use **HTML, CSS, and JavaScript** for styling and functionality
• Implement modular React components

### API Integration

Use the **GitHub REST API v3** to fetch developer profile data and repository information.

Students should ensure proper handling of:

• Loading states
• API request errors
• Rate limiting from the GitHub API

### Free Development Tools

Students can complete the project using the following free tools:

• **VS Code** for development
• **GitHub** for version control and repository hosting
• **Vercel or Netlify** for frontend deployment

These platforms offer free tiers suitable for this project.

# Bonus Challenges (Optional)

Students who want to extend their project can implement additional features such as:

- Compare multiple GitHub profiles
- Visualize contribution activity using a heatmap
- Add repository search or filtering options
- Export profile analysis as a PDF report
- Add sorting options for repositories

These features help demonstrate advanced frontend development and data visualization skills.

# Submission Guidelines

To receive the **Project Completion E-Certificate**, students must submit the following:

**Required**

• GitHub repository link containing the complete project source code
• Live deployed project link

Students can deploy their project using free platforms such as **Vercel or Netlify**.

**Optional (Recommended)**

• 2–3 screenshots demonstrating the application interface and features

All submissions must be made through the **official Scoop Labs Project Lab submission form**.

Submission Form:
*https://scooplabs.in/project-lab*

Ensure that your GitHub repository includes a **README file explaining the project and setup instructions**.

# Certification

Students who successfully complete and submit their project will receive a **Project Completion E-Certificate from Scoop Labs**, an **Authorized Training Partner of the National Skill Development Corporation (NSDC)** under the **Skill India Mission**.

This certification recognizes the successful completion of an industry-oriented development project.

# Project Showcase Opportunity

Selected high-quality submissions may be **featured in the Scoop Labs Project Gallery**.

Projects showcased in the gallery serve as a **public portfolio reference**, providing additional credibility when students share their work with recruiters and potential employers.

# Need Help?

If you have questions or face technical issues while building the project, you can contact the Scoop Labs team for guidance.

Support Email:
**info@scooplabs.in**

# Scoop Labs Project Lab

Building **industry-ready projects for aspiring developers.**

Scoop Labs is an **Authorized Training Partner of the National Skill Development Corporation (NSDC)** under the **Skill India Mission**, committed to helping learners develop practical technical skills through real-world projects.