LEAP 4.0 DOC Weeks 1

Business Requirements Document

Project Topic: GIT Extractor

Developers: Tribe G

Document Date: 26-06-2021

Target Date: 17/07/21

1.1 Problem Statement

An Application Programming Interface (API) is a tool set that programmers can use in helping them create software. A good API will have clear and concise commands that a programmer can use and reuse, so they don't have to build everything over again.

Let's take an example Why we need API :-

When we want to retrieve some data from a website, say a list of new items from any website. Suppose there were no APIs, your developer wants to go to the website, then manually do the scrapping and complicate with the data. This will consume a lot of time which actually effect the app development; instead he gets focused on data. The similar applies for retrieving data from a Database. Now suppose there is an API, the developer can retrieve the data easily, or even update the data without downloading entire data to any application backend. Simply an API helps the developer to think less on the data and concentrate more on the app development and front end.

1.2 Objective

We know from experience that business objectives tend to vary, but generally they center around ways to:

- Improve customer experience and loyalty
- Bring new innovative business models
- Build a platform ecosystem
- Increase revenue channels
- Build a platform ecosystem
- Increase partnership with other players in the ecosystem

2. Approach

- API framework is created by using ReactJS and use Git API for fetching data from Git servers.
- To identify user request and list Repositories or user information which will be unique in each case can be achieved by using Axios library.
- To show the result of the search using Axios library, JSON (JavaScript Object Notation) is used.
- Firebase is used for the backend.
- Netlify is used for the deployment of the application's it provides optimized and fast cloud computing services.

3. Technical & Non-Technical Requirements

3.1 Technical Requirements

ID	Requirements
3.1	Technical Requirement 1. Languages to be used- JavaScript, CSS, HTML 2. Framework- ReactJS 3. Database- Firebase 4. Deployment- Netlify.

3.2 Non-Technical Requirements

ID	Requirements
3.2	Non-Technical Requirement • Documentation which will include BRD (Business Requirement Document) and solution document. • Design layout and working • Testing and presenting live demo of API after development to the stakeholder

3.3 Non-Functional Requirements

ID	Requirements
3.3	Non-Functional Requirement • Dependencies- Use of Netlify and Github pages will give a broader spectrum of opportunities to use this API from anywhere by anyone. • Efficient- Making an interface which is organized and developed conscientiously as per client's requirement with touches of optimizations for optimal experience. • Speed- Providing the interface with minimum wait time and maximum efficiency making the system smooth, fast and reliable

3.4 User Requirements

ID	Requirements
3.4	User Requirement • User Id and password is required to access the website. • To fetch details user has to have GitHub Id of the party they require details of.

3.5 Security Requirements

ID	Requirements
3.5	Security Requirement : • If new user wants to use our application, they have to Signup first. • Passwords should be 6 characters long. • Feature like forgot password will help you retrieve the account if you forget the password

4.Scope of the Project

- This project focuses on giving client a user friendly environment and provides an aid for things like finding profiles and there directory.

- Due to pandemic there are a lot of coders sitting out there with an amazing repository and still unseen our tribe would like to give them spotlight and recognition.

- There a lot of students and amateurs who want to find codes for their assignments, mini projects, so on and so forth but GitHub are too complex to use and finding directories is a pain.

- Github extractor can be used to find profiles which are suited for the client's projects and hence can recruit or contact them in a much easier fashion.

5. Use Case

5.1 Use Case

Actors (Internal Users, External Users, System)	Any user can use this application
Description	The use case describes the fetching details of the username from the Git account .It contains various steps that involves the authentication of the user, verification of username and showing and generating the report of the user in file format.
Preconditions	The user must have an account to use this application. If they are new to the website they can create their new account and login.
Triggers	The need and use of codes and repository in an easy to access fashion
Flow of events	1. The user first login to the application. 2. Enter the username, if the username is not valid it shows Invalid username message. 3. For valid user API fetches the details of the user such as bio, Repository, date of creation and last update of repository. 4. We can also generate Report and download it. This report also contains hyperlink. 5. At lasteach search or fetching instruction is shown as acknowledgement.
Success Criteria	The fetching request is successful and client gain the user info.