

## Synopsis

### Title

SoPRO (Social PROgrammer)

### The Problem Statement

Programmers who are learning a new skill or working on a project may face difficulty during their work and may not have someone in their knowledge who's familiar with that technology. So we will make a good learning platform where programmers around the globe would be able to collaborate and share their own codes.

### Reason

To create a social platform for learners that'll help resolve queries or get new ideas. Existing platforms coding forums have a certain issues such as :-

- **Letting new users face the entire community at once-**

A new user who just started and needs help with his coding might get lost in the vast platform. Search results may be irrelevant and unhelpful at times.

SoPRO will tackle this by sorting the posts language-wise and letting the user see the content that is meant for him.

- **Many people don't get helped-**

It is likely that an error faced by a programmer has already been seen and solved by another programmer on such platforms but the various parameters of a program may be different, their causes may be different as each person has their own way of coding.

SoPRO tackles this by helping programmers find issues in his OWN personal code rather than somebody else's similar code.

- **Beginners may get disappointed-**

Often it is seen that when a new contributor joins such a community he faces various issues such as being downvoted for no reason.

SoPRO tackles this with a transparent medium where there's no such concept of a downvote as 'EACH ANSWER MATTERS'.

### Objectives

- To create a social platform for programmers.
- To help a programmer debug and resolve his code errors.
- To give personalised solutions to errors.
- To share new ideas and thoughts on coding.
- To make a feed that is relevant to a particular skill.
- To make it available on multiple platforms.

## Feasibility Study

- **Technical Feasibility:** We can say that our Application would be technically feasible because we are not getting any difficult related to resource of development and maintenance of the project. It's a Application that requires manually maintaining the databases with the zip files; by the Developer. It is easy to use for all, and everyone can contribute here.
- **Economically Feasibility:** The project that we are building would be economical as it wouldn't be taking any extra tools other than required tools for development and is available for free. We have made sure that neither user nor the developer need to spend any money for the accessing and development of the system. Everything will be done using Open Source tools available.
- **Schedule Feasibility:** It is defined as the state of being probable and will be completed within the scheduled time. We are working from day one on the project.
- **Operational Feasibility:** It is related to the measurement of the performance of the system for which purpose it was developed. We can say that our project will be operationally feasible as we are using RESTful API, to Get,Put,Post and delete the data.

## Future Scope

Scope for extension into a major project:

- GLA could use soPro for all the coding contests held in college by various clubs,we could provide a portal for that in soPro.
- Implementing "File Search" feature for quick searching of particular files in large and complex structures.
- A PC version of soPro.
- Serverless Architecture for deploying soPro
- Monolithic Architecture for building soPro from scratch

## Methodology:

We are using Microservice Architecture to provide the Independency of using a tech stack to each member, which would ensure a great environment where everyone would be choosing their preferred language and tools.

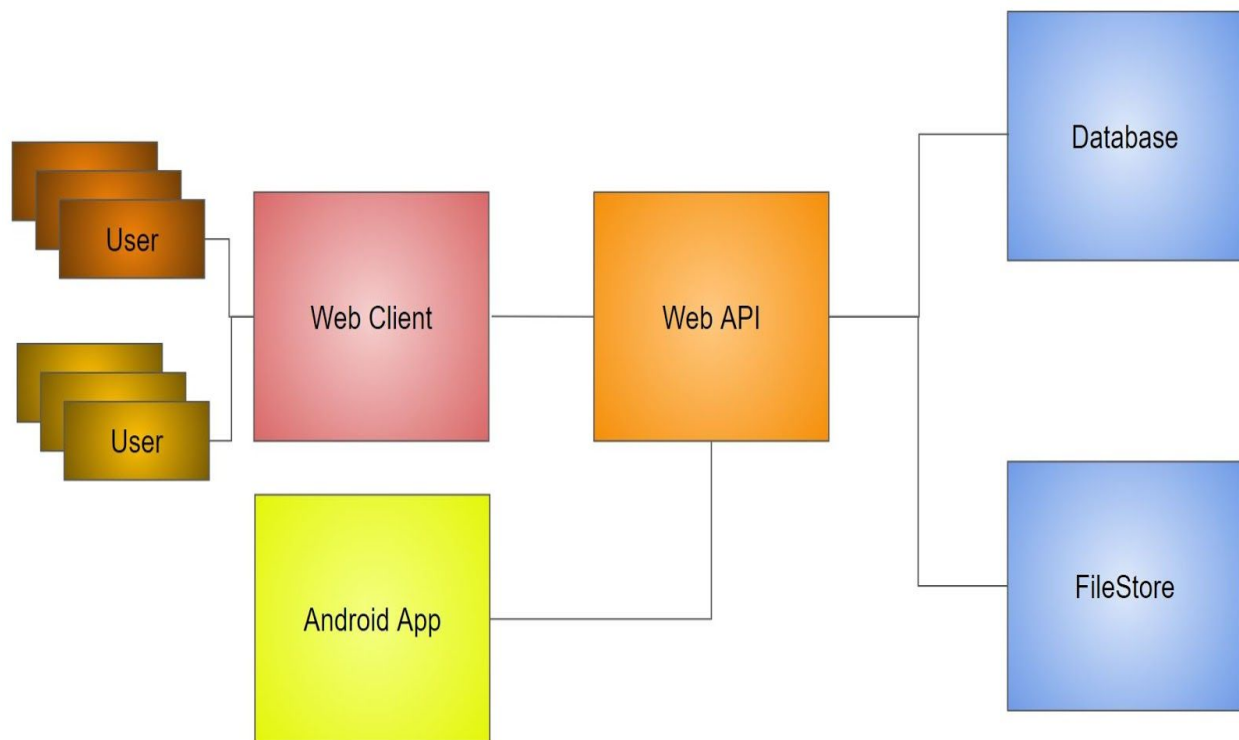
## What are microservices?

Microservices -- also known as the microservice architecture - is an architectural style that structures an application as a collection of services that are

- Highly maintainable and testable
- Loosely coupled
- Independently deployable
- Organized around business capabilities
- Owned by a small team

The microservice architecture enables the rapid, frequent, and reliable delivery of large, complex applications. It also enables an organization to evolve its technology stack.

## Summary:



We divided the whole into the following services:

- Web Client
- Web API
- Database
- FileStore
- Android App

### ***Web Client:***

WebClient would consist of

- Backend will be the logical side of the WebClient which will help to provide the dynamic nature of WebClient and would make requests to the API according to the Users action.
- FrontEnd will be the visual side of the WebClient of the microservice project, it will provide an interface for users to login/register and use various functionality.

### ***Web API:(RESTful)***

Application Programming Interface(API) will receive requests from Client server, process them and return a response depending on the nature of the request made by the client.

API would be using bcrypt as a password-hashing function followed by Salting the hashes to save in user login credentials inside databases so not even the developer could access a user account with login credential inside the database.

The API will be implemented with an Authorization method to check whether the User trying to make a request is authorized with a valid JWT(JSON Web Token) token or not and return a response accordingly.

### ***Database:***

The Web Dev team decided to use a Non-Relational Database (MongoDB) so that Database migration couldn't be the issue which is quite normal for Relational Database, decreasing the time duration between development and production, as the project is in its initial stage.

### ***FileStore:***

Saving an Image/File in a database may not be efficient therefore we would be using AWS S3 for storing the Images, ZIP .

### ***Android App:***

Our team will also provide Android applications for cross-platform functioning, resulting in a larger user base for creating communities. Thus, increasing the number of problem solvers and decreasing waiting time.

### **Software used:**

- Notepad
- Visual Studio Code
- Google Chrome
- Postman
- MongoDB Compass
- Android studio

### **Testing technologies:**

- POSTMAN

### **What contribution does the project make and where? :**

It feels frustrating and infuriating when a few lines of code makes you stop coding ? Isn't so. Well if this is the way for experienced coders think about the newbies who when trying new algorithms get stuck and are not able to get solutions. Many might argue go on Google for this but for beginners it is overwhelming to go through all the stuff.

For tackling this problem, our platform will provide you with active coders who prefer the same language as the user does. This will not only help in getting the solutions for as small as a doubt the user has. This way every doubt will be solved.

### **Bibliography:**

1. <https://microservices.io/>
2. <https://www.mongodb.com/>
3. <https://restfulapi.net/>
4. <https://www.postman.com/>
5. <https://jwt.io/>
6. <https://www.npmjs.com/package/bcrypt>

7. <https://aws.amazon.com/>
8. <https://aws.amazon.com/elasticbeanstalk/>
9. <https://aws.amazon.com/s3/>