#### Developing Platform-independent Resuable Functional Components

Brief Description:

No matter what kinds of software, e.g., web-based, IOS&Android or Applets, all of them have some common ground. From a business perspective, they all have some common business components, such as user management, SSO (Single Sign On), authentication and verification, permission management, reporting and analytics, payment processing, localization and Internationalization and many others.

From a developer perspective, all kind of systems should have some common features, such as environment agnostic which means these components can communicate with different kinds system and deploy on various cloud-platform. Other functional requirements like CI/CD pipeline, performance monitor, defining custom metrics, API generated, traffic control etc.

On top of these features, The business logic component can be developed quickly and with high quality, so no matter the development is done by a team within a company or a freelancer, this skeleton system can help them shorten development time. Which

Through out this project, I will dedicate myself to developing such a skeleton system with multiple reusable and platform-independent compoenents. This also caters to the current popular agile development model.

Where is the challenge located within such a system?

* The project covers broad technical and business knowledge, which means I need to learn a lot of things.
* Developed components need to integrate within other systems smoothly, regardless of the development language or platform
* Making the system independent of deployment platform is technically challenging.

**Requirements of the project**

Technical Specification:

Essential requirements:

* User management(SSO/Oauth)
* Permission management(Data & Pages permission)
* Integrate with third-party technic for log searching/ metrics monitor, CI/CD etc
* Containerized these components and Used K8S to autoscale
* CI/CD and API generating
* All components should be scalable, high availability
* Build a home photo-sharing system on top of these basic components.

Recommended requirements:

* Build a serverless local GP system
* Combine these components with the serverless local GP system

Optional requirements:

* Re-developing these components but in a serverless way(E.g., AWS serverless)