

Presents

Hackathon AFourathon 3.0 - 2023

Hi There,  
  
Welcome to the AFourathon 3.0 Hackathon!

Before you begin working on your project, please ensure that you read through and work with the **Digital Asset Engineering Guide**. It will provide valuable information about the requirements of your project and the criteria for evaluation. We highly recommend that you keep it handy on both days - 1st Demo Video Submission, and Final Presentation Day.

**You are required to record a 15 min Demo Video & it should include the following:**

* Discuss and Showcase the Overall Architecture: 2 Min
* Discuss and Showcase the Current Features: 1 Min
* Discuss the Future Scope or Plan for the Project: 2 min
* Main Project Demo (Technologies Used, Assessment Criteria Covered & Working Solution): 10 Min

**Note:** On the final presentation day, you must follow the same drill and showcase your project to the AFourathon Judges Panel.

-------------------------------------------------------------------------------------------------------------------------------------

AFour Digital Asset Design, Engineering and Support Guide

-------------------------------------------------------------------------------------------------------------------------------------

# Solution

Solution Purpose

<Describe the pain areas that we are planning to address with this asset. Who are intended end users/customers.>

Scope and No Scope

<List the key use cases and also list under no scope dependencies from the third party services or libraries>

# Design

UX Design

UX Diagrams

<Provide URL for UX diagrams>

UX Design Principles/Guidelines/Standards

<Describe the principles used in designing the UX e. G. interactive design, material design, etc. Describe any new ideas, disruptive ideas used in design>

Database Design

<Add Database schema and other supporting documents or urls for supporting documents.>

Database Design Principles/Guidelines/Standards

<Mention any new ideas or database design principles/standards that are used. E. g. De-normalized some of the information for efficiency.>

API Design

<Provide list of APIs and purpose of each API. API Swagger URL and other supporting documents.>

<Provide API Design and Development Principles/Guidelines/Standards. E.g. Used SOLID principles, used Java coding standards as per –URL–->

Technology Architecture Design

<Provide Technology Component Diagram (FE, BE, Database, Cache, Messaging, etc.)>

<Provide Deployment Architecture Diagram (Containers, Clusters, etc.)>

<Mention Tools and Technologies>

<Describe Scalability Features. E. g. Containerized deployment, Elastic Load Balancing, etc.>

<Describe High Availability Features. E. g. Multizonal deployment, Hybrid cloud deployment, etc. Provide high availability value e.g. 99.99% or 99.999%, etc.>

<Describe Security Features. E. g. Integration with third party IAM, RBAC system, etc.>

<Describe Logging / Auditing Features. E. g. Date/Time, User columns for audit trail in each table, >

<Describe Monitoring Features>

<Notification Features>

# Development

Source Code/Git

<Provide list of repositories and their purpose - e. G. front end code, back end code, other business components>

Branching strategy

<Describe branching strategy>

Setting up Developer Machine

<List instructions to set up developer machine. If lengthy point to the URL for the same>

Unit Testing Method

<Mention unit testing method. E. g. Peer developer/lead will write the unit test cases which the developer first will convert into unit test code and then will write application code to pass all the unit test cases.>

# Testing

Test Scope

<Mention scope of testing. This can also be repeated in test strategy/test plan document>

BVT Test Cases

<Provide list of BVT test cases>

Test Automation Objective & Scope

<Provide test automation goals and scope of test automation>

Test Strategy / Test Plan

<Provide Test Strategy, Test Plan URLs>

Test Automation Strategy / Test Automation Plan / Test Automation Design

<Provide Test Automation Strategy/ Test Automation Plan/ Test Automation Design URLs>

Test Automation Branching Strategy

<Describe test automation branching strategy. E. g. Test automation branching will be inline with application/component source code. For every release of application/component there will be a release of test automation build to keep automation code maintenance to the minimum.>

Test Code/Git

<Provide URL(s) of Test Automation code>

# DevOps

Dev Integration Environment Set up

<Provide infrastructure provisioning steps>

<Provide step by step instructions of CI+CD for setting up Dev integration Environment>

QA Environment Set up

<Provide infrastructure provisioning steps>

<Provide step by step instructions of CI+CD for setting up QA Environment>

Staging Environment Set up

<Provide infrastructure provisioning steps>

<Provide step by step instructions of CI+CD for setting up Staging Environment>

<Provide steps to back up and restore staging data for staging environment. Provide steps for sensitive and PII production data masking.>

# Production Deployment

Infrastructure

<Mention step by step instructions for infrastructure setup>

Application

<Mention step by step instructions for application deployment and configuration>

Logging / Auditing Configurations

<Mention configuration required for enabling logging, auditing.>

Scalability Configurations

<Mention configuration for scalability>

Security Configurations

<Mention configurations for enabling security features.>

Credentials

<Mention BU and IT Points of contact for Credentials. Mention frequency to change the credentials.>

Certificates

<Mention certificate name, certificate issuing authority, mention location to download the certificate and steps to install the certificate.>

# Production Testing

<List of important points to be verified on production environment

E. g. Configuration points,

Key Use cases>

# Backup and Recovery

<Provide steps for database backup and recovery>

<Provide steps for messaging component recovery>

<Provide steps for services component recovery>

<Provide steps for front end application component recovery>

# Monitoring

<List of key parameters to monitor.>

<List of emails, aliases to notify the health status.>

# Routine Maintenance

<Provide areas for routine maintenance and describe step by step instructions for the same

E. g. Access Key Rotation

Service/Size Limit Evaluation

Certificates Renewal

Third party libraries renewal

Third party services renewal

Other dependencies renewal (e. G. Renewal of generic components/services developed internally)

>

# Emergency Maintenance

<Provide emergency maintenance details

E. g. Point of Deployment Contacts

E. g. Point of Engineering Contacts

E. g. Support Channels>

<Provide list of emergency use cases and troubleshooting hints for each case>

Emergency Use Case 1 and Troubleshooting

Emergency Use Case 2 and Troubleshooting

Emergency Use Case 3 and Troubleshooting

# Marketing Support

<Mention the URL of the Application/Asset Overview document.>



AFour Technologies Pvt. Ltd. Office 501, 5th Floor, Sterling Tower,  
Pan Card Club Road, Baner, Pune 411045, India