Task:
Provide a Test Plan that describes an approach on how to test this system: types, levels of testing, tech stack, tooling.

1. Goals

Goal	Description
Early Feedback	We want to know as early as possible about the code (we are writing) will work or not .
High Modifiability	We want to make sure it is easier for other developers to contribute from the same team or different team without being bothered about breaking it. Allowing us to make changes at faster pace and evolve constantly.
Faster Release Cycles	Shorten the test and regression cycle and move from manual validation to automated validation test.

2. Approach

All testing tasks will be conducted in line with the Software Test Life Cycle (STLC) and in support of the Software Development Life Cycle (SDLC). The documents used within the SDLC will be completed both by the Test Team and the project participants that are responsible for providing information and deliverables to the Test Team.

Incremental testing is used in agile development methods and hence, every release of the project is tested thoroughly. This ensures that any bugs in the system are fixed before the next release.

Functional testing will be provided for API including smoke, regression, automation testing. Due to Microservices guidelines, also contract testing will be done.

3. Test Levels/Types

Each Microservice must have test coverage for:

Test Type	Description/Metrics	Responsible
Unit tests	Unit test coverage must be 70%, however if it's any lower than it must increase by at least 5% in each iteration	**To be done by Dev Team**
Contract Test	Boundary values, Equivalence classes, Mandatory and optional inputs /outputs	**To be done by Manual Team**
Component Test	Positive flows and Business exceptions	**To be done by Manual and Automation Team**
Integration Tests	Verify interfaces for external sources consumed by microservice in test	**To be done by Automation Team**
End to End test (UI testing)	Test Service as blackbox from end user / consumer points of view	**To be done by Manual and Automation Team**

4. Tech Stack

Tech stack used for automation framework	
Java 11	OOP language
JUnit 5	Test runner
Wiremock	for stubbing out external services
AllureReport 2.0 Framework	Test report tool
Log4j	A flexible lightweight logger
REST-assured	for writing REST API-driven end-to-end tests
Maven	Build tool
Tech stack used for development	
Java 11	OOP language

Kubernetes	Enable automated scalability while utilizing the minimum amount of cloud resources which manage docker containers.
Kafka	Distributed event streaming platform

5. Tooling

Platform	Description
Jira	Issue tracking product developed by that allows bug tracking and agile project management.
Confluence	Web-based corporate wiki
GitLab	Web-based DevOps lifecycle tool that provides a Git-repository manager providing wiki, issue-tracking and continuous integration and deployment pipeline features
Postman, Soap UI	API client that makes it easy for developers to create, share, test and document APIs
IntelliJIDEA	Integrated development environment