unext

Internship as a Service

~Crafting Your Professional Path

Internship as a service – Technologies covered

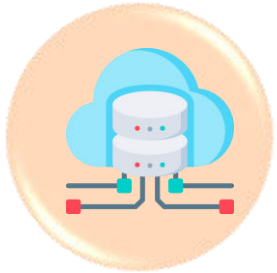
Full Stack Programs

DevOps

Cybersecurity

Automation Testing

Generative AI



Cloud

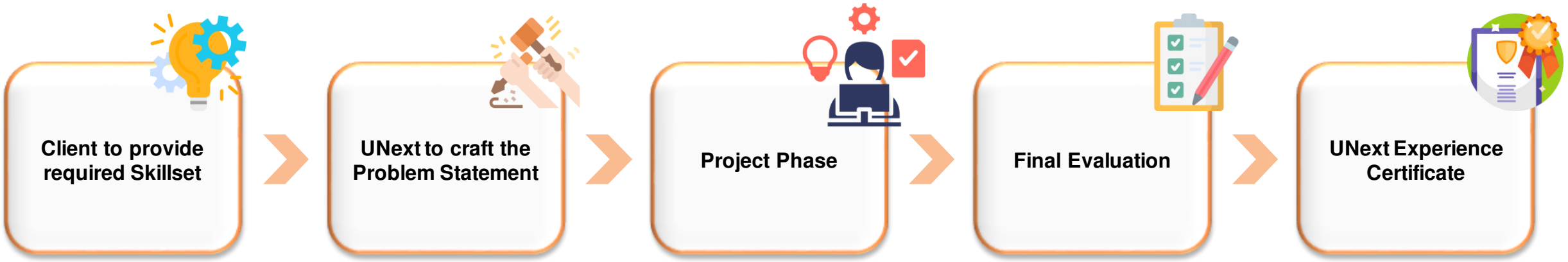
Big Data

Data Analytics

AI/ML

The content and the program approach is strictly confidential. It is strictly forbidden to share any part of this program design approach with any third party

Internship as a service – Delivery Flow



12 weeks (12 sprints) each sprint shall have epic goals

Milestones set every week to assess participant's progress

Mentors assigned to guide participants throughout the project

The content and the program approach is strictly confidential. It is strictly forbidden to share any part of this program design approach with any third party

Internship as a service - Competencies

Pre - requisites

Foundation

Track	Git/GitHub	Java	Maven
		Python	pip
		GO	modules
		NodeJS	npm

Database		Backend	Frontend
SQL	NOSQL	SpringBoot	ReactJS
		Django	Angular
		ExpressJS	Vue
		Laravel	ReactJS
		Gin	Flutter

With Prompt Engg – Github Copilot / Tabnine / AWS etc

Competencies

NFR	Design Document, Response Time, Page view time, Throughput, Availability, Concurrent users, CI/CD Pipelines, Release wise pipeline configuration, Failure alerts, Error Handling for Microservices, Logging and Monitoring, High availability 99.99% , Testing of HA with relevant tools
Security	Secure Access, Secure Data Access, Security Standards (OWASP / GDPR),Handling PII Data
Accessibility	Responsive web design, accessibility design , Error Page Management, Email Notifications for Server Downtime, Interoperability
Volumes	Test data preparation and management
Engineering Practices	100% Method Coverage, 95% Block/Line Coverage, Code Quality, Code Complexity, CPU & Memory Profiling, Unit Test Reports, Monitoring & Alerting, Release Cadence, Feature Release, Subsequent Bug-Fix Release, Subsequent Prod Release, Subsequent Patch Release (if any), E2E testing, BDD Approach for testing
Cloud	Deployment, https / ssl , DNS Management
Usability	Intuitive & Accessibility, SEO compliant, RWD, Localization, WCAG 2.x ,Offline Functionality

- **Participants are expected to have the pre-requisite skills**
- **A pre-assessment based on the Pre-requisite skill sets will be carried out and participants who don't meet the required criteria will be suggested to take a refresher course of 5 days duration.**

The content and the program approach is strictly confidential. It is strictly forbidden to share any part of this program design approach with any third party

Learner Journey (1/2)

PRE-ASSESSMENT

Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
Epic Goals To setup the development environment, for backend and front-end services	**Epic Goals** To build the secure and sustainable application which can be scalable with cloud formation & follow engineering principles	**Epic Goals** Optimize application performance for faster load times, efficient resource utilization.	**Epic Goals** Set up automated deployment pipelines for CI/CD to streamline the release process.	**Epic Goals** Set up automated deployment pipelines for CI/CD to streamline the release process with cloud	**Epic Goals** Implement security measures such as encryption, data validation, and protection against threats. And working with the system capability checks
Functional Delivery (Includes FE/BE) Setup application with HelloWorld application	**Functional Delivery (Includes FE/BE)** Build functional stories BE/FE Build application with TDD first approach with relevant tools	**Functional Delivery (Includes FE/BE)** Setting up application with caching Building application user stories FE/BE	**Functional Delivery (Includes FE/BE)** Building application user stories FE/BE Building unit test cases and analysing the reports to pass the build with CI/CD	**Functional Delivery (Includes FE/BE)** Building application user stories FE/BE	**Functional Delivery (Includes FE/BE)** Building application user stories FE/BE
Non-Functional Delivery Scaffold the project, with proper folder structure (for FE/BE) + best practices. Containerize the application which is scafolded	**Non-Functional Delivery** 100% Method Coverage 95% Block/Line Coverage Code Quality Code Complexity Setup cloud(to be discussed) architecture	**Non-Functional Delivery** Testing application with load testing stress testing, regression, sanity testing	**Non-Functional Delivery** Setting up quality gateway with tools like SonarQube to setup CI/CD pipeline Parameterizing the application for dev/text/prod branch Creating appropriate tags for the build when fails	**Non-Functional Delivery** Release wise pipeline configuration. Failure alerts Error Handling for Microservices	**Non-Functional Delivery** Secure Access (Authentication and Authorization) Secure Data Access Handling PII Data

Scheduled Mentor Connect : Mentor Guidance every week (4-6 hours per week)

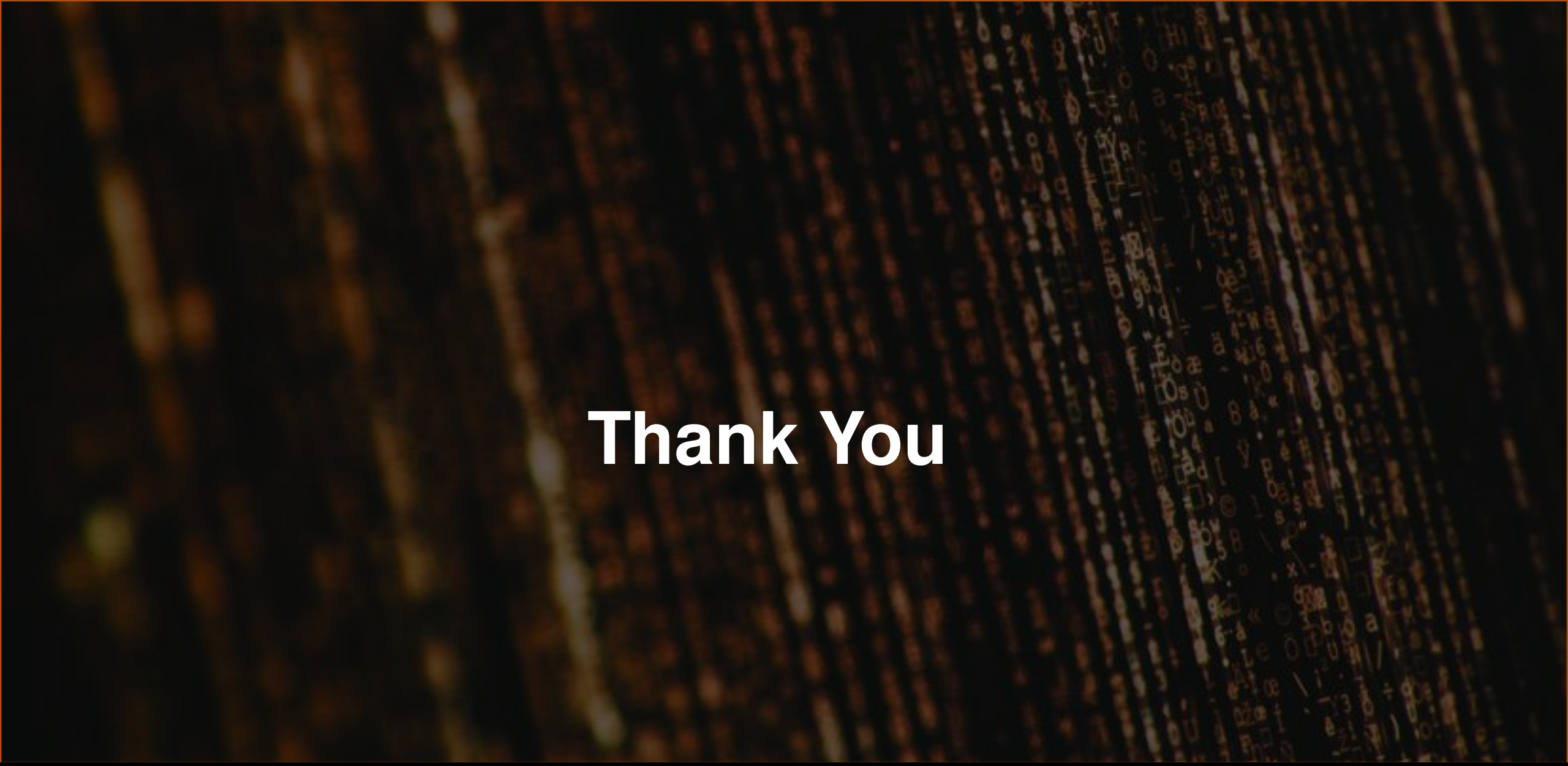
The content and the program approach is strictly confidential. It is strictly forbidden to share any part of this program design approach with any third party

Learner Journey (2/2)

Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
Epic Goals	**Epic Goals**	**Epic Goals**	**Epic Goals**	**Epic Goals**	**Epic Goals**
Implement security measures such as encryption, data validation, and protection against threats. And working with the system capability checks	Ensure the application adheres to accessibility standards for users with disabilities follow WCAG 2.x	Implement backup and recovery to safeguard user data against loss or corruption.	Implement monitoring and logging solutions for tracking application health and performance.	Ensure compliance with relevant regulations and standards regarding user data privacy and security	Bugfixes, working on release cadence, release notes documentation (application, user documentation)
Functional Delivery (Includes FE/BE)	**Functional Delivery (Includes FE/BE)**	**Functional Delivery (Includes FE/BE)**	**Functional Delivery (Includes FE/BE)**	**Functional Delivery (Includes FE/BE)**	**Functional Delivery (Includes FE/BE)**
Building application user stories FE/BE	Building application user stories FE/BE	Building application user stories FE/BE	Building application user stories FE/BE	NA	NA
Non-Functional Delivery	**Non-Functional Delivery**	**Non-Functional Delivery**	**Non-Functional Delivery**	**Non-Functional Delivery**	**Non-Functional Delivery**
Testing the data with volumes (min 1 million) CPU & Memory Profiling Monitoring & Alerting	Responsive web design Accessibility design Error Page Management Email Notifications for Server Downtime Interoperability Intuitive & accessibility SEO Compliant , RWD Localization / work with i8n standards	Setting up backup mechanism for database with repl factors Working with application to scale instances more than 1, ensuring there is no downtime. Delayed backup	Trigger the build failure on communication. Failure alerts Error Handling for Microservices Logging and Monitoring High availability 99.99% Testing of HA with relevant tools	Security Standards (OWASP / GDPR) Handling PII Data	Release Cadence Feature Release Subsequent Bug-Fix Release Subsequent Prod Release Subsequent Patch Release (if any)

Scheduled Mentor Connect : Mentor Guidance every week (4-6 hours per week)

The content and the program approach is strictly confidential. It is strictly forbidden to share any part of this program design approach with any third party


Thank You