YIP Yebox Internship Program

Training Curriculum

2023.

FRONTEND DEVELOPMENT

Week 1

Day 1: Introduction to Frontend Architecture

- Understanding the role of frontend architecture in web development.
- Exploring different architectural patterns (e.g., MVC, MVVM, Flux, Redux).
- Best practices for structuring front-end projects.

Day 2: Introduction to React.js

- Overview of React.js and its core concepts.
- Setting up a basic React project using Create React App.
- Components and JSX syntax.

Day 3: State Management with Redux

- Understanding the need for state management in complex applications.
- Setting up Redux in a React project.
- Actions, reducers, and the Redux store.

Day 4: Advanced React Components

- Container and Presentational components.
- React Hooks and their role in modern React development.
- Building reusable and maintainable components.

Day 5: Unit Testing in React

- Introduction to testing libraries like Jest and Enzyme.
- Writing unit tests for React components and Redux actions/reducers.
- Test-driven development (TDD) approach.

Weekend Break: Encourage participants to review the topics covered during the week and work on a small project to reinforce their learning.

Week 2:

Advanced React and Performance Optimization

Day 6: Advanced State Management

- Using middleware in Redux.
- Asynchronous actions with Redux Thunk and Redux Saga.

- Handling complex state scenarios.

Day 7: React Router and Navigation

- Implementing client-side routing using React Router.
- Nested routes and route guarding.
- Handling dynamic routes and query parameters.

Day 8: Performance Optimization in React

- Identifying performance bottlenecks in React applications.
- Strategies for performance optimization (e.g., code splitting, lazy loading).
- React.memo() and useMemo/useCallback hooks.

Day 9: Error Handling and Logging

- Implementing error boundaries to handle runtime errors gracefully.
- Integrating logging mechanisms to track errors in production.
- Strategies for debugging React applications effectively.

Day 10: Introduction to End-to-End Testing

- Understanding end-to-end (E2E) testing and its importance.
- Setting up E2E testing using tools like Cypress or Selenium.
- Writing E2E tests to cover critical user flows.

Weekend Break: Participants can use this time to work on a larger project incorporating the concepts learned so far and preparing for the next phase.

Week 3-8: build season

BACKEND DEVELOPMENT

Week 1:

Advanced Node.js and Backend Architecture

Day 1: Introduction to Advanced Node.js

- Overview of Node.js and its asynchronous, event-driven nature.
- Advanced features in Node.js (streams, clusters, worker threads).
- Setting up a Node.js project and organizing the file structure.

Day 2: Asynchronous Programming in Node.js

- Understanding callbacks, promises, and async/await in Node.js.
- Error handling in asynchronous code.
- Using the Node.js built-in 'util' module for promisify and error handling.

Day 3: Building RESTful APIs with Express

- Recap of REST architecture and principles.
- Creating a RESTful API using Express.js.
- Middleware and request processing in Express.

Day 4: Data Storage and Databases

- Different database options for Node.js (e.g., MongoDB, PostgreSQL).
- Integrating a database into a Node.js project.
- Implementing CRUD operations with a database.

Day 5: Authentication and Authorization

- Implementing user authentication with JSON Web Tokens (JWT).
- Role-based access control (RBAC) and middleware for authorization.
- Securing API endpoints.

Weekend Break: Participants can review the topics covered during the week and practice building RESTful APIs with various database systems.

Week 2:

Testing, Performance, and Real-time Communication

Day 6: Unit Testing and Test Automation

- Introduction to testing frameworks (e.g., Jest, Mocha).
- Writing unit tests for Node.js backend code.
- Setting up test automation for continuous integration.

Day 7: API Testing and Validation

- Using tools like Postman or Supertest for API testing.
- Validating API responses and status codes.
- Integration testing with test databases.

Day 8: Performance and Scalability

- Identifying performance bottlenecks in Node.js applications.
- Caching strategies using Redis for performance optimization.
- Scaling Node.js applications with load balancing.

Day 9: WebSockets and Real-time Communication

- Understanding real-time communication and use cases.
- Implementing WebSockets in Node.js with libraries like Socket.io.
- Building real-time features (e.g., chat, notifications).

Day 10: Security Best Practices

- Common security vulnerabilities in Node.js applications.
- Applying security best practices, such as input validation and sanitation.
- Using helmet.js for HTTP headers security.

Weekend Break: Participants can use this time to work on a larger project incorporating the concepts learned so far and preparing for the build phase.

CLOUD ENGINEERING

Week 1:

Introduction to DevOps and Version Control

Day 1: Understanding DevOps

- What is DevOps and its principles.
- Benefits of implementing DevOps in software development.
- Overview of DevOps tools and practices.

Day 2: Version Control with Git

- Introduction to Git and its importance in DevOps.
- Setting up a Git repository and basic version control operations.
- Branching and merging strategies.

Day 3: Collaborative Development with Git

- Working with remote repositories (e.g., GitHub, GitLab).
- Collaborating with teammates using pull requests.

- Code reviews and best practices.

Day 4: Continuous Integration (CI) with Jenkins

- Introduction to Continuous Integration (CI) and its role in DevOps.
- Setting up Jenkins for automating CI/CD pipelines.
- Building, testing, and deploying applications using Jenkins.

Day 5: CI/CD Pipeline with Jenkins

- Creating a complete CI/CD pipeline with Jenkins.
- Integrating automated testing and deployment stages.
- Handling notifications and reporting in the pipeline.

Weekend Break: Participants can review the topics covered during the week and explore more CI/CD tools like Travis CI, CircleCI, or GitLab CI/CD.

Week 2:

Infrastructure as Code (IaC) and Cloud Deployment

Day 6: Infrastructure as Code (IaC) with Terraform

- Introduction to Infrastructure as Code (IaC).
- Setting up Terraform for automating infrastructure provisioning.
- Writing Terraform configurations to manage cloud resources.

Day 7: Configuration Management with Ansible

- Understanding the role of configuration management in DevOps.
- Installing and configuring Ansible.
- Creating playbooks to manage application configurations.

Day 8: Docker Containers and Container Orchestration

- Introduction to Docker and containerization.
- Building Docker images and running containers.
- Container orchestration with Kubernetes.

Day 9: Deploying Applications on Cloud Platforms

- Deploying applications on cloud platforms like AWS, Azure, or Google Cloud.
- Leveraging cloud services for scalability and reliability.
- Managing infrastructure and applications in the cloud.

Day 10: Monitoring and Logging in DevOps

- Implementing monitoring and alerting for applications.
- Using tools like Prometheus, Grafana, and ELK stack for logging and monitoring.
- Understanding performance and availability metrics.

Weekend Break: Participants can use this time to work on a DevOps project that incorporates the concepts learned during the previous week and prepares for the final week.

PRODUCT DESIGN

Week 1 - 2

- Introduction to design
- Good designs vs Bad designs
- Categories of design
- Fundamental design principles (C.R.A.P)
- Ideation & How to solve problems
- Introduction to product design

- Design thinking process
- Planning user research and interviews
- Understanding users & what they want
- How to approach an interviewee
- How to create questionnaires & surveys
- Knowing & applying user persona
- UI vs UX and who a UI/UX designer is and what they do
- Introduction to UI design tool (figma)

- Sprint planning
- What is storyboarding
- Wireframes

- Low-fi design
- Mid-fi design
- High-fi design
- Component and design element

- Introduction to design system
- Create style guide
- Preparation for your first design project
- Finding real life problem
- Proposing solution to the problem

Week 3 - 8: Practical

PRODUCT MANAGEMENT

Week 1 - 2
Product Management Basics

1. Introduction to Product Management
2. PM Required Competencies
3. Product Management vs Project Management
4. Phases of Project Management
5. Phases of Product Management Life cycle
6. Communicating with a Product Requirement Document -
    - What is a PRD
    - Breaking down a PRD
    - Personas
    - User scenarios & storytelling
    - Defining Requirements/Features

The Basics of Product Development

- From Idea to Action
- The Concept of Minimum Viable Product in Product Development
- Feature Prioritization Concepts
- Working With Design
- Working With Engineers

The Project Process

- Project Initiation
- Project Planning
- Project Execution
- Intro to Agile Methodologies (Scrum, Kanban)
- Project Closure
- Project Management Tools

Go to Market

- Testing
- Product launch planning
- Listen to the Market - Customers

Week 3 - 8: Practical

MOBILE APP DEVELOPMENT

Week 1 - 2

The Leap of Faith

Introduction to Mobile App Development

- What are mobile apps?
- Differences, similarities and examples of mobile platforms.
- What's Mobile App Development and types of Mobile Apps.
- Mobile App Development best practicies.

What is Flutter? and Why Flutter?

- Introduction to Dart and Dartpad
- Why Flutter Uses Dart.
- Dart basis.
- The anatomy of a Flutter App.

Writing your first Flutter App

- Prerequisites of Flutter Development.
- Environment setup
    - Flutter SDK installation
    - Android studio installation
    - The Android simulator
    - For Mac - xCode installation
    - For Mac - Homebrew installation
    - For Mac -xCode command line tools installation
    - For Mac - Cocopods installation
    - For Mac - The iOS simulator
- Running your firs Flutter App - The Counter App.
- The Counter App breakdown and explanation

Creating Flutter Apps from Scrartch

- The concept of widgets
- Types of widgets
- Scaffolding a Flutter App
- Working with Assets

Building UI's in Flutter

- Creating a BMI Calculator App

Resources:
What is Flutter - By Flutter Dev Team,
What is flutter? Amazon
Dart Basis,
Dart Basis 2,
Dart full crash course
Android Studio, VS Code
Widgets from Google,
Life of a Widget?, Flutter Widgets - Youtube

Building UI

Building stunning UIs in Flutter

- Building an E-Commerce App
- Navigating across pages with MaterialPageRoute

Building stunning UIs in Flutter contd

- Building a ToDo list App..
- Navigating across pages with MaterialPageRoute contd
- Temporary database with Maps and Lists

Building stunning UIs in Flutter contd

- Building a Movie App.
- Exploring Flutter and Dart packages and plugins
- Introduction to logging and debugging.

Building complex Realworld UIs in Flutter

- Building a Twitter Clone.
- Introduction to statemanagement.
- Introduction to Flutter Bloc.

Building complex Realworld UIs in Flutter contd

- Building a Twitter Clone contd.
- Introduction to Atomic design principles.
- Introduction to Resource Locators - eg. getIt.

Resources:
Complete Flutter UI Design - Complete UI design in Flutter
Build your first Flutter App - Flutter Codelab - Beginer Flutter Codelab 2
Flutter UI basis - Flutter E Commerce App with Firebase

Week 3: Networking - Building connected Apps

Introducing Firebase

- Retriving and Storing data in Firebase
- Creating a picture collage App

Introducing Firebase

- Creating a connected ToDo list App
- Firebase Auth

Introduction to Rest APIs

- Building a Food Recipe App
- Types of Rest API requests
- The Http and Dio package
- Parsing JSON data

Resources:
Fetch data from the internet, Network Request
Making authenticated requests
Sending data to the internet
Updating data over the internet
Delete data on the internet
Parsing JSON
Persist data with SQLite
Reading and writing files
Store key-value data on disk

Networking and Database

What's a database

- Introducing Share Preferences, Flutter secured storage, Flutter hive
- Reading and writing files to local storage - Persisting data with Share Preferences

Week 3 - 8 : Practical

- Using TheMealDB API, build a functional food recipe App