

TechShots



## Contents

1. What we are
2. Why we are
3. What we offer
4. How it will help students

## **What we are**

We are starting TechShots, a community of programmers.

## **Why we are**

We aim to help students grow by helping them in frameworks and technologies used by most of the companies. The idea is to give them a flavor of what is next in their post-college life by making them implement problems that are being solved in an organization. Further helping the students decide what they want to choose as expertise in their profession be it **android, ios, web development or backend development.**

About us: We have mentors with working experience with companies like **Hotstar, Practo, UrbanClap, 1mg, Times Internet, Goibibo.** All mentors have good knowledge of data structure and algorithm with good experience in competitive programming

## **What we offer**

1. Free live tutorial on various programming languages and framework with doubt session.
2. College sessions, meetups, medium articles for the same.
3. Quiz and tests for interview preparation.
4. Chance to be a part of a community where you can learn and teach.

## **Session topics -**

1. Android App development
2. iOS App development
3. Python programming language
4. Backend in NodeJS
5. Web development
6. Introduction to AWS
7. Firebase
8. C/C++/JAVA
9. Data structures and algorithms
10. Aptitude preparation
11. Basic devops and system design fundamentals

We need students to bring their laptops.If college could provide internet connection and projectors that will be great.

## **Day 1 Briefing -**

- Intro to techshots
- Brief intro to how industry operates
- intro to different software developer career option kuch aisa

Next session on specific tech stack will be based on student selection

We will also select some students and try to help them get internships and offer in organizations where mentors can refer them.

## Android App development

1. Fundamentals of mobile development
2. Java and Kotlin
3. Android Components in Depth part-1
4. Android Components in Depth part-2
5. Getting data from network
6. Getting data from storage
7. Understanding Android OS
8. Advance android views
9. Design pattern in android
10. Android Architecture components
11. MVVM vs MVP vs MVI with test cases
12. Dependency injection using dagger 2
13. Rx java and Kotlin Coroutine
14. Project part -1
15. Project part -2
16. Project part -3

Project will include making a sample using proper architecture and production ready code that will be reviewed by company professional mentors(more than one).

1. Fundamentals of mobile development
  - a. Getting familiar with mobile development, its basics and how is it different from other platforms.
  - b. Tools and framework needed
2. Java and Kotlin
  - a. various programming paradigm
  - b. functional vs object oriented
  - c. basics of java
  - d. basics of kotlin
3. Android Components in Depth part-1,part-2
  - a. Activity
  - b. Fragment
  - c. Views
  - d. Services
  - e. Broadcasts
  - f. Intent
  - g. Parcel
  - h. making UI with xml files
4. Getting data from network
  - a. HttpURLConnection
  - b. volley

- c. retrofit
  - d. gson
- 5. Getting data from storage
  - a. Database
  - b. shared preference
  - c. file
- 6. Understanding Android OS
  - a. OS architecture
  - b. How an app actually runs in Android
  - c. How OS and application interacts
  - d. JIT vs AOT
  - e. DVM vs JVM
- 7. Advance android views
  - a. view lifecycle
  - b. custom views
  - c. animation
- 8. Design pattern in android
  - a. Some basic design pattern like builder, factory, observer etc that can help you write clean code
- 9. Android Architecture components
  - a. Live Data
  - b. ViewModel
  - c. ROOM
- 10. MVVM vs MVP vs MVI
  - a. Testable Code
  - b. Discussion on basic sample code written using all three architecture
  - c. Writing test cases
- 11. Dependency injection using dagger 2
  - a. Basics of Components, Modules
  - b. Dagger with android
  - c. Introduction to component dependency and subcomponents.
- 12. Rx java and Kotlin Coroutine
  - a. Introduction to Observer pattern
  - b. building blocks of RX - Observables, Flowables, subjects
  - c. Practical examples of Rx
  - d. Introduction to coroutines

## iOS App Development -

1. Introduction to iOS Apps
  1. what they are ?
  2. Mobile development Paradigms
2. Xcode Setup and Basic Tutorial
3. Swift vs Objective C
4. Swift Basics
  1. Collections and Operators & control flow
  2. Optionals & Properties & closures
  3. Value types & reference types (Class, Struct, Enum)
  4. Memory Management
  5. Protocols ,Extensions
  6. Access Control and Error Handling
5. Introduction to UIKit
  1. Interface builder & Storyboards & Xibs
  2. ViewControllers
    - Viewcontroller lifecycle
    - Purpose of view controller
  3. Views
    - View life Cycle
    - Purpose of views
  4. Navigation controller
  5. Tab bar controller
  6. UIScrollView
  7. UITableView
  8. UICollectionView
  9. UIStackViews
  10. Introduction to Animations
  11. Playing them all together
6. Introduction to URL Session
  1. How to use purpose and example
  2. Codable
7. Introduction to Data Storage
  1. UserDefaults
  2. KeyedArchiver
  3. CoreData
  4. Disk Storage
  5. Data Storage with Example
8. Grand Central Dispatch and Operations
9. Notifications
10. Architecture
  1. Understanding how iOS App works
  2. Run Loop
  3. Apple's MVC and Today's MVC
  4. MVVM and MVP
  5. VIPER
11. Design Patterns(Observer, Decorator, builder and many more)

12. SOLID Principles
13. More in depth Protocol Oriented Programming and Generics
14. Dependency Injection
15. REXSwift
16. CocoaPods and PodFile
17. Projects
  1. Project 1 and make it as CocoaPod
  2. Project 2 build project and use that Project 1 CocoaPods
18. Introduction to AVKit
19. Other UIComponents and UIGestures
20. Unit Test Cases

## **Backend development with NodeJS, Express and MongoDB**

1. Javascript basics
2. Understanding NodeJS
3. Design patterns in NodeJS
4. Working with Express
5. MongoDB basics
6. Working with MongoDB
7. Authentication and Authorization
8. Unit tests
9. Project

Project will include making a sample using proper architecture and production ready code that will be reviewed by company professional mentors(more than one).

1. Javascript basics
  - a. Data types, operators, loops etc
  - b. Closures
2. Understanding Node js
  - a. Event loop
  - b. Async tasks
  - c. Node js modules, npm
3. Design patterns in Node js
  - a. Callbacks, Promises, EventEmitter
  - b. Factories
  - c. Strategy pattern
4. Working with Express
  - a. How express works
  - b. Setting up basic http apis
  - c. Express middleware
5. MongoDB basics
  - a. What is mongodb - documents, collections
  - b. NoSQL queries
  - c. Indexing



6. Working with MongoDB
  - a. Node with mongodb using Mongoose
  - b. Defining schemas/models
  - c. Pre/post hooks
7. Authentication and Authorization
  - a. Setting up auth using JWT tokens
  - b. Setting up authorization based on user access
8. Unit tests
  - a. Writing unit tests using Mocha, chai and sinon

## **Data Structures and Algorithms**

### **Data Structures**

1. Array
2. Linked Lists
3. Stacks
4. Queues
5. Binary Trees
6. Binary Search Trees
7. HashMap
8. Heap
9. Graph
10. Matrix
11. Advanced Data Structures

### **Algorithms**

1. Analysis of Algorithms
2. Searching
3. Sorting
4. Greedy Algorithms
5. Dynamic Programming
6. Pattern Searching
7. Other String Algorithms
8. Backtracking
9. Divide and Conquer
10. Geometric Algorithms
11. Mathematical Algorithms
12. Bit Algorithms
13. Graph Algorithms
14. Randomized Algorithms
15. Branch and Bound

**Classes specific to various programming languages can also be included depending on the number of students interested in them.**

**A special class for aptitude will also be included open for everyone.**