


**Finite Loop Club**

NMAM Institute of Technology, Nitte, SH1, Karkala, Karnataka, 574110  
+91 8197903771 ♦ [finiteloopclub@gmail.com](mailto:finiteloopclub@gmail.com) ♦ <https://finiteloop.co.in/>


## WEB DEVELOPMENT ROADMAP

### FUNDAMENTALS

#### 1. Basics


-  How does the internet work? (Full Course)

#### 2. HTML, CSS

-  HTML & CSS Full Course for free  (2023)

Learn HTML and CSS together don't need to learn everything you can visit back anytime

#### 3. Javascript

- [Learn JavaScript](#) (Recommended)
-  Build 15 JavaScript Projects - Vanilla JavaScript Course

### Frontend Development:


Don't start learning frameworks without having solid foundation on HTML CSS and JS

**JS Framework:** (Choose any one React/Svelte/Vue/Angular/..)

#### 1. React: (Recommended)


-  React JS Full Course - Build 4 Projects in 5 Hours | Zero to Hero

#### 2.CSS:

-  Tailwind CSS Full Course for Beginners | Complete All-in-One Tutorial | 3 Hours

Learn along with learning metaframeworks

#### 3. NextJS: (Metaframework)

-  Next.js 14 Complete Course 2024 | Next.js 14 Full Stack App Tutorial for Beginners

#### 4.Additional (optional):

1. [Three.js](#) (Advanced 3D Library)
2. [WebGL](#) (Low Level Graphics Library)
3. [GSAP](#) (CSS Animation Library)

To keep yourself up to date with the Web development trend

- [Fireship - YouTube](#)
- [Awesome - YouTube](#)

More resources:

- [Frontend Mentor | Challenges](#) (HTML CSS Projects)
- [Kevin Powel](#) (CSS)
- [freeCodeCamp.org - YouTube](#)
- [Lama Dev - YouTube](#) (React and Next Projects)
- [MDN Web Docs](#)

## Backend Development:

### 1. Programming language (Choose any one)

1. [Javascript](#) (Recommended)
2. [Python](#)
3. If you have learnt metaframeworks in the frontend development then no need to learn any specific backend framework

### 2. Backend Frameworks(Choose any one):

Python:

1. [Django](#) (Recommended)
2. [FastAPI](#)


Javascript:

1. [Node and Express](#) (Highly recommended for full stack developer)
2. [Bun and elysia](#)

### 3. Choose database

1. [Postgres](#) (Recommended)
2. [MongoDB](#)
3. [SQLite](#)

### 4. Backend as service framework

1. [Learn Firebase](#) (Firebase) (Recommended)
2.  [Supabase Crash Course](#) (Supabase)

### 5. Cloud Services

1. [AWS](#) (Recommended)
2. [Google Cloud](#)
3. [Azure](#)

# APP DEVELOPMENT ROADMAP

## Java

- **BEGINNER** - [Beginner Android Programming - Caleb Curry](#)
- **INTERMEDIATE** - [Android Development Full Course - freeCodeCamp.org](#)
- **ADVANCED** - [Complete Android Development Playlist](#)

## Kotlin

- **BEGINNER** - [Beginner Kotlin Course](#)
- **INTERMEDIATE** - [Build and Publish an Android App - Full Course with Kotlin](#)
- **ADVANCED** - [Complete Android Development with Kotlin Course - freeCodeCamp.org](#)

## Flutter

- **BEGINNER** - [Oneshot Flutter Beginner Course](#)
- **INTERMEDIATE** - [Flutter Course by FreeCodeCamp](#)
- **ADVANCED** - [Complete Flutter Course](#)

# BLOCKCHAIN/ WEB3 ROADMAP

## BASICS

**1. Getting started with Blockchain** (What is Blockchain, history of Blockchain, Bitcoin White Paper, Blockchain Applications)

- [All about Blockchain | Simply Explained](#)

**2. Blockchain Fundamentals Introduction** (Nodes, Hashing, Cryptography, Digital Signature, Consensus)

- [Web3 Basics](#)

**3. Solidity Basics** (Use cases, Contract structures)

- [Learn Solidity in 20 Minutes!](#)

## INTERMEDIATE

**4. Smart Contracts Development** (Remix, Foundry)

- [Learn Solidity, Blockchain Development, & Smart Contracts | Powered By AI - Full Co...](#)

**5. Ethereum Learning Track**

- [Ethereum Learning Track](#)

**6. PROJECT** (Build Blockchain app using Solidity smart contracts and Ethereum)

- [Build Your First Blockchain App Using Ethereum Smart Contracts and Solidity](#)

## ADVANCE

**7. [Mantle Learning Track](#)** (**Layer 2 Blockchain** is a off-chain network, system, or technology built on top of a blockchain to help extend its capabilities such as higher transaction throughputs, lower transaction fees)

**8. [DeFi Specialization](#)** (Introduction and code implementation of DEX, DAI Projects)

**9. [Dao Specialization](#)** (Decentralised Autonomous Organisation)

**10. PROJECT** [Build and Deploy a Modern Web 3.0 Blockchain App | Solidity, Smart Contracts, Crypto](#)

# AI & ML ROADMAP

## Foundation:

### 1. Mathematics:

- a. **Linear Algebra:** Vectors, matrices, matrix multiplication, eigenvalues, and eigenvectors.
- b. **Calculus:** Derivatives, integrals, partial derivatives basics.
- c. **Probability and Statistics:** Basic probability theory, random variables, probability distributions, Bayesian inference.

### 2. Programming:

- a. **Python:** Basic syntax, data structures (lists, dictionaries), loops, conditionals, functions.
- b. **Libraries:** Numpy, Pandas, Matplotlib, Scikit-learn.

## Core Machine Learning:

1. **Concepts:** Supervised vs. unsupervised learning, training vs. testing, overfitting vs. underfitting.

### 2. Algorithms:

- a. **Supervised Learning:** Linear regression, logistic regression, decision trees, k-nearest neighbors (KNN), support vector machines (SVM).
- b. **Unsupervised Learning:** K-means clustering, hierarchical clustering, principal component analysis (PCA).

### 3. Model Evaluation and Validation:

- a. **Metrics:** Accuracy, precision, recall, F1 score, ROC-AUC.
- b. **Techniques:** Cross-validation, train-test split, confusion matrix.

### 4. Data Preprocessing:

- a. **Data Cleaning:** Handling missing values, outliers.
- b. **Feature Engineering:** Feature scaling (normalization, standardization), feature selection.

## Intermediate Topics:

### 1. Advanced Machine Learning Algorithms:

- a. **Ensemble Methods:** Random forests, gradient boosting, XGBoost.
- b. **Neural Networks and Deep Learning:** Basics of neural networks, activation functions, backpropagation.

### 2. Specialized Language Libraries:

- a. **TensorFlow/PyTorch:** Introduction to deep learning frameworks, building and training neural networks.

- b. **Keras:** High-level neural networks API. Part of tensorflow library.

## Project Areas:




1. **Kaggle Competitions:** Participate in machine learning competitions to apply your skills.
2. **Personal Projects:** Build and deploy your own machine learning models.
3. **Topics for projects:**
  - a. **Natural Language Processing (NLP):** Text preprocessing, sentiment analysis, language models.
  - b. **Computer Vision:** Image classification, object detection, convolutional neural networks (CNNs).
  - c. **Reinforcement Learning:** Basic concepts, Q-learning

## Advanced Topics:

1. **Advanced Deep Learning:**
  - a. **Recurrent Neural Networks (RNNs):** LSTM, GRU.
  - b. **Generative Adversarial Networks (GANs):** Understanding and implementing GANs.
  - c. **Convolutional Neural Networks(CNNs):** Used for image classification and computer vision tasks.
2. **Optimization and Deployment:**
  - a. **Model Optimization:** Hyperparameter tuning, model pruning.
  - b. **Deployment:** Model serving, cloud platforms (AWS, GCP, Azure), Docker.

## Resources:

1. **Online Courses:**
  - a. **Coursera:** Coursera courses offered by Andrew Ng. You can get these courses for free by applying through financial aid.
    - i. [Machine Learning Specialization \[3 courses\] \(Stanford\) | Coursera](#)
    - ii. [Deep Learning Specialization \[5 courses\] \(DeepLearning.AI\) | Coursera](#)
    - iii. For people who want to go deeper into the mathematics: [Mathematics for Machine Learning and Data Science Specialization \[3 courses\] \(DeepLearning.AI\) | Coursera](#)
  - b. **Edx:** Best for beginners and learning faster.
    - i. [CS50's Introduction to Artificial Intelligence with Python | Harvard University](#)

- c. **Youtube:** There are many tutorials available at your fingertips. These are just some of the famous ones. If these don't match your tastes please do refer other creators. There are some phenomenal resources for AI and ML on youtube. For math do checkout [3b1b](#) youtube channel. Phenomenal content for those who are interested in deeper understanding.
  - i. [codebasics - YouTube](#) - Best resource for beginners.
  - ii. [CampusX - YouTube](#)
  - iii.  Artificial Intelligence For Beginners | AI Tutorial For Beginners | Intelli...
  - iv.   Artificial Intelligence Full Course | Artificial Intelligence Tutorial For...

## 2. Practice Platforms:

- a. **Kaggle:** Competitions, datasets, and notebooks.
- b. **GitHub:** Explore open-source projects and contribute. With the rise of LLMs , the drive towards open source projects in AIML have skyrocketed. More and more people are building models and uploading them to github. Reading them will help you understand how things are written and understand how concepts are applied in the real world from theory to practice.

***For people who don't have dedicated gpu for training heavy models, or just want to collaborate with other people, do checkout [Google Colab](#), A Free GPU access for training models. It is the best place to write and experiment with machine learning and train faster with the help of dedicated GPUs.***

## VERSION CONTROL (Git&GitHub)

- [GIT CHEAT SHEET](#)
- [Git and GitHub for Poets - YouTube](#)(Best out there)
- [Complete Git and GitHub Tutorial](#)
- [Git and GitHub for Beginners - Crash Course](#)