



Presidential Initiative for Artificial Intelligence and Computing (PIAIC)

<https://www.piaic.org>

Certified Web 3.0 and Metaverse Developer

Including Cloud-Native and Ambient Computing/IoT Specializations

A One Year Panaverse DAO Program

<https://www.panaverse.co>

Version 3.0.0 - January 2023

The internet is undoubtedly the most significant technological development in human history. Although the industry has progressed immensely since its inception, we are just scratching the surface of what it can do for us. Web3 and metaverse technologies improve the internet as we know it today by adding innovative new features and enhancements. The metaverse reimagines the front end and Web 3.0 reimagines the back end of the web. Web 3.0 and the Metaverse will work together and complement each other. Whereas Web 3.0 is primarily concerned with who will own and rule the internet of the future, the metaverse concept is concerned with how users will interact with it. There will be two kinds of Metaverses. The Virtual Metaverse will be for short-duration uses, while the Augmented Metaverse will revolutionize society.

Citi is the latest Wall Street firm to issue a bullish forecast for Web 3.0 and the Metaverse, which describe a future internet vision based on decentralized technologies and virtual worlds. Citi said in a [research report published in March 2022](#) that the metaverse economy could have a total addressable market of up to \$13 trillion and can capture five billion users by 2030.

The participants can choose from these three specializations:

- I. Web 3.0 and Metaverse Specialization
- II. Cloud-Native Computing Specialization
- III. Ambient Computing and IoT Specialization

This program is designed for beginner-level students who will start by learning software development from scratch. The first two quarters are common in all specializations and focus on learning Object-Oriented Programming and cutting-edge Full-Stack Web 2.0 development.

After completing the first two quarters the students will select one or more specializations mentioned above.

This Web 3.0 and Metaverse specialization focuses on developing full-stack Web 3.0 and Metaverse experiences for the next generation of the internet by specializing in building worlds that merge the best of cutting-edge decentralized distributed blockchains with 3D metaverse client experiences.

The Cloud-Native Computing Specialization focuses on Containers, Kubernetes, and CDK for Kubernetes.

The Ambient Computing and IoT Specialization focuses on building Smart Homes, Offices, Factories, and Cities.

The Outcome for Participants of the Program

After completing this program students will be able to develop 2D and 3D planet-scale experiences that scale instantly to support even the most demanding applications, while at the same time delivering up to 90% cost savings. This stack which will be covered in this program is ideal for a broad set of applications. For example, enterprises that have hundreds of thousands of applications, or software as a service (SaaS) vendors that have multi-tenant environments with hundreds or thousands of databases, can use this stack across the entire enterprise.

The students who graduate from this one-year program will be able to earn \$50 per hour (\$96,000 per annum), to begin with. On one hand, this will be a great opportunity for students and professionals of Pakistan and on the other, it will provide a much-needed boost to the economy by increasing software exports.

1. Top 5 'Metaverse' jobs that will rule the future of tech industry
<https://content.techgig.com/hiring/top-5-metaverse-jobs-that-will-rule-the-future-of-tech-industry/articleshow/87581325.cms>
2. Blockchain Developer Salary - Jun 2022
<https://web3.career/web3-salaries/blockchain-developer>
3. Web3 Salaries Soar to \$750,000 for Rank-and-File Devs
<https://thedefiant.io/web3-soaring-salaries/>
4. The Metaverse, Blockchain Gaming, and NFTs: Navigating the Internet's Uncharted Waters
<https://newzoo.com/insights/trend-reports/newzoo-report-on-metaverse-blockchain-gaming-nft-2022>
5. How To Become Metaverse Developer: Scope, Skills, and Salary
<https://www.blockchain-council.org/metaverse/how-to-become-metaverse-developer/>

The Plan in a Nutshell: Earn While You Learn

We aim to train one million developers across Pakistan. These developers will be experts in cutting-edge Web 2.0, Web 3.0, and Metaverse Technologies. Given the current financial situation in the country, we are combining our teaching and dollar-making business activities. This is going to be a totally new kind of program where students will be taught how to earn dollars.

Grading:

Students will be graded based on a percentile grading system.

<https://en.wikipedia.org/wiki/Percentile>

https://en.wikipedia.org/wiki/Percentile_rank

A-Grade: 78 - 99 Percentile

B-Grade: 41 - 77 Percentile

C-Grade: 23 - 40 Percentile

D-Grade: 1 - 22 Percentile

F-Grade: Anyone who doesn't appear in two or more exams

Note: Anyone who receives an F-Grade will be removed from the program. Students who receive a D-Grade will be put on probation, and be required to earn a grade of C or above in the next quarter, to remain in the program. Anyone absent from an exam will be deemed to have received a score of zero.

Important Note: If a PIAIC candidate doesn't appear in a Quiz at the scheduled time announced by management 10% score will be deducted from the test score for every week of delay.

Program of Studies

This is a year-long online program consisting of four quarters (*see details below*) and will mainly be focused on hands-on learning by training students to develop projects. The classes will mainly be taught on the weekends or after 6:00 pm (Pakistan Time) on weekdays so that everyone can attend. This will be a hybrid teaching model comprising main onsite classes and supplemented by online Zoom labs as well as recorded videos.

YouTube Live Channel: <https://www.youtube.com/@panaverse/streams>

Github Repos: <https://github.com/panaverse>

Discord: <https://discord.gg/qMd63rS6>

Facebook: <https://www.facebook.com/groups/panaverse>

Twitter: https://twitter.com/Panaverse_edu

Course Syllabus

Quarter I

CS-101: Programming and Web User Experience Fundamentals

Duration: 13 Weeks

Course Description:

We will start the program by learn the fundamentals of Object-Oriented programming using JavaScript and TypeScript. We will also understand the latest Web trends i.e. Web 3.0 and Metaverse concepts and try to understand their working from the perspective of the users.

Course Outline:

1. HTML and CSS (Homework)

Learn HTML by Hira Khan (Watch Recorded Videos)

<https://www.youtube.com/playlist?list=PLKvqnz8z1zWQ3BALy86tIXICkG874wAc6>

Learn CSS Intro by Hira Khan (Watch Recorded Videos)

https://www.youtube.com/playlist?list=PLKvqnz8z1zWQSWIen_zUSEBmtqzPLuRob

2. Web 3.0 and Metaverse Theory

Introduction to Panaverse DAO

<https://docs.google.com/presentation/d/12C1s4UBTIR9nZNEkRE6aAuGeRA3s92jx-8nJKo9jdH4/edit?usp=sharing>

Web 3.0 User Guide

https://docs.google.com/presentation/d/1F5br9aIwO0-fmZHqWy_eHO2N_jwJLmQCy4cG8rd4ctw/edit?usp=sharing

Complete Web 3 Assignments included in the Web 3 User Guide

Virtual and Augmented Metaverse User Guide

https://docs.google.com/presentation/d/1ADk87hQ0Etr2PfmN9XH7TQ0CHl6XSP_7JWNUbzPdDNc/edit?usp=sharing

3. Fundamentals of JavaScript ([ECMAScript 2022 Language Specification](#))

Learn JavaScript by Zeeshan Hanif (Watch Recorded Videos)

<https://www.youtube.com/playlist?list=PLKvqnz8z1zWQdc0NSLknxmxBch5gLOqyo>

Chapters 2-6, 13 of JavaScript from Beginner to Professional: Learn JavaScript quickly by building fun, interactive, and dynamic web apps, games, and pages

https://www.amazon.com/JavaScript-Beginner-Professional-building-interactive/dp/1800562527/ref=sr_1_4

JavaScript Book Code

<https://github.com/PacktPublishing/JavaScript-from-Beginner-to-Professional>

Getting Started Exercises with JavaScript and Node.js

<https://github.com/panaverse/typescript-node-projects/blob/main/getting-started-exercises.md>

Fundamentals of JavaScript and Node.js Quiz

Topics Covered in the Quiz:

- A. Background of JavaScript and How to use JavaScript in Browser
- B. Variables, Primitive data types Analyzing and modifying data types, and Operators (Chapter 2 of JavaScript from Beginner to Professional)
- C. Intro to Node.js, .mjs files, Modules, NPM, import, export, and using external modules with npm:
https://docs.google.com/presentation/d/1eOLJTN0FyVAw6lMUvA_XYLORKnudWXyzXsB6sg5oPIs/edit?usp=s_haring
- D. How to accept user input in your Node.js JavaScript programs, this will allow us to create interactive Node.js console programs using prompt-sync library. The last example in this presentation shows you how to use prompt-sync library in your Node.js programs:
https://docs.google.com/presentation/d/1eOLJTN0FyVAw6lMUvA_XYLORKnudWXyzXsB6sg5oPIs/edit?usp=s_haring
Note: After this point, we will do all our class work and exercises using Node.js in .mjs files. We will also be able to develop interactive Node.js console programs which will greatly help the students to learn.
- E. Using Arrays and Objects in Node.js Programs (chapter 3 of JavaScript from Beginner to Professional)
- F. Using if and if else statements, else if statements, Conditional ternary operators, and switch statements in Node.js programs (chapter 4 of JavaScript from Beginner to Professional)
- G. Using while loop, do while loop, for loop, for in, and for of loop in Node.js (chapter 5 of JavaScript from Beginner to Professional)
- H. Using Basic functions, Function arguments, Return, Variable scope in functions, Recursive functions, Nested functions, Anonymous functions, and Function callbacks in Node.js (chapter 6 of JavaScript from Beginner to Professional)
- I. Using Concurrency, Callbacks, Promises, async / await, and Event loop (chapter 13 of JavaScript from Beginner to Professional)
- J. JavaScript promises, mastering the asynchronous
<https://www.codingame.com/playgrounds/347/javascript-promises-mastering-the-asynchronous/what-is-asynchronous-in-javascript>
- K. New JavaScript Features in ECMAScript 2022 and 2021
<https://dev.to/brayanarrieta/new-javascript-features-ecmascript-2022-with-examples-4nhg>

4. Object-Oriented Programming with TypeScript

Chapters 1-11 of Learning TypeScript: Enhance Your Web Development Skills Using Type-Safe JavaScript

https://www.amazon.com/Learning-TypeScript-Development-Type-Safe-JavaScript/dp/1098110331/ref=sr_1

Learning Repository

<https://github.com/panaverse/learn-typescript>

In Class Companion projects and articles for Learning TypeScript

<https://www.learningtypescript.com/>

Homework Project

<https://github.com/panaverse/typescript-node-projects>

Fundamentals of TypeScript Quiz

TypeScript Professional Proficiency Quiz

5. TypeScript for React

Minimal TypeScript Crash Course For React
With Interactive Code Exercises

<https://profy.dev/article/react-typescript>

6. Fundamentals of Version Control with Git (Recorded Videos)

Version Control using Git and GitHub By Sir Zeeshan Hanif (Watch Recorded Videos)

https://www.youtube.com/playlist?list=PLKueo-cldy_HjRnPUl4G3pWHS7FREaizF

Chapters 1, 2, 3, and 4 Learn Version Control with Git: A step-by-step course for the complete beginner by Tobias Günther

We will also cover these readings:

<https://help.github.com/articles/markdown-basics/>

<http://stackoverflow.com/questions/5009600/difference-between-fork-and-branch-on-github>

<http://stackoverflow.com/questions/3329943/git-branch-fork-fetch-merge-rebase-and-clone-what-are-the-differences>

<https://git-scm.com/book/en/v2/Git-Branching-Rebasing>

<http://git-scm.com/book/en/v2/Git-Branching-Remote-Branches#Tracking-Branches>

For practice: <https://try.github.io/levels/1/challenges/1>

Homework:

<https://www.datacamp.com/courses/introduction-to-git-for-data-science>

Git Cheatsheet

<https://www.kdnuggets.com/2022/11/git-data-science-cheatsheet.html>

Git Quiz before the start of Quarter 2

Total Questions: 60, Total Time: 75 minutes

Note: Git study material and videos are being made available in the first quarter so that students are able to use Git immediately. The Git Quiz will be conducted in the first week of the next quarter i.e. second quarter and not in the first quarter.

Quarter II

W2-201: Developing Planet-Scale Web 2.0 Cloud Applications

Duration: 13 Weeks

Course Description:

The objective of this course is to teach participants to develop customer-facing planet-scale Websites, Full-Stack Apps and templates, Dashboards, and Multi-Cloud Serverless APIs. By the end of quarter the participants will be able to develop and deploy web platforms like Facebook, Shopify, etc. The technologies covered in this course will include Next.js 13, Figma, Tailwind CSS, Chakra UI, tRPC, GraphQL, Prisma, Cockroachdb Serverless (PostgreSQL Compatible), AWS Serverless Technologies, and Cloud Development Kit for Terraform (CDKTF).

Course Outline:

1. React Crash Course (Homework)

React JS Full Course 2022 | Build an App and Master React in 1 Hour

<https://www.youtube.com/watch?v=b9eMGE7QtTk>

2. Next.js 13 Web Development

Next 13 Official Documentation

<https://beta.nextjs.org/docs>

Learn Next.js 13 Learning Repo

<https://github.com/panaverse/learn-nextjs>

3. APIs with Next.js and tRPC

Build a tRPC CRUD API Example with Next.js

<https://codevoweb.com/build-a-trpc-crud-api-example-with-next-js/>

Stop building REST APIs for your Next.js apps, use tRPC instead

<https://brockherion.dev/blog/posts/stop-building-rest-apis-for-your-next-apps/>

4. SQL and Prisma

Start from scratch with relational databases

<https://www.prisma.io/docs/getting-started/setup-prisma/start-from-scratch/relational-databases-typescript-postgres>

SQL For Beginners Video Tutorial

<https://www.youtube.com/watch?v=5hzZtqCNQKk>

Database Management Systems and SQL – Tutorial for Beginners

<https://www.freecodecamp.org/news/dbms-and-sql-basics/>

5. Next.js 13 using Chakra UI (Remote Zoom Class)

CSS Flexbox Explained – Complete Guide to Flexible Containers and Flex Items

<https://www.freecodecamp.org/news/css-flexbox-complete-guide/>

6. Next.js 13 using TailwindCSS (Remote Zoom Class)

Modern CSS with Tailwind, Second Edition by Noel Rappin
<https://pragprog.com/titles/tailwind2/modern-css-with-tailwind-second-edition/>

7. UI/UX Design with Figma, TailwindCSS and Chakra UI (Remote Zoom Class)

Designing and Prototyping Interfaces with Figma: Learn essential UX/UI design principles by creating interactive prototypes for mobile, tablet, and desktop by Fabio Staiano

https://www.amazon.com/Designing-Prototyping-Interfaces-Figma-interactive/dp/180056418X/ref=sr_1_1_sspa

Figma Design Kit for TailwindCSS

<https://www.figma.com/community/file/768809027799962739>

Chakra UI Figma Kit

<https://www.figma.com/community/file/971408767069651759>

8. AWS Application Composer (Remote Zoom Class)

What is AWS Application Composer?

https://www.youtube.com/watch?v=BujE_tik5r8

Event driven apps with AWS Application Composer

<https://www.youtube.com/watch?v=p411uh363jQ>

Visually design and build serverless applications quickly

<https://aws.amazon.com/application-composer/>

Must Have: Create Free AWS Account

AWS Free Tier <https://aws.amazon.com/free/>

Note: For AWS Free Tier you will need a credit or debit card. The easiest way for Pakistani students is to open a bank account with Meezan Bank. Open a Meezan Aasan Account if you have no earning proof. Meezan Bank will charge Rs. 1100 for ATM Debit Card and there is a requirement of Rs. 100 minimum deposit. The card will be delivered in a few weeks. Our students have reported that this Debit card works with AWS.

Now you can create a virtual VISA Debit card through the NayaPay app (<https://www.nayapay.com/>) which can be used to create AWS/GCP/Azure accounts. This is good news for those students who find it difficult to create physical debit and credit cards from banks.

You can also get a \$300 credit

https://pages.awscloud.com/adoptf90d_GLOBAL_POC-credits.html

9. Multi-Cloud GraphQL Serverless API Development with Cloud Development Kit for Terraform (CDKTF) (Remote Zoom Class)

Web 2.0 Projects

1. **Next.js Projects**
<https://github.com/panaverse/nextjs-projects>
2. **Styling Next.js Projects using TailwindCSS and Chakra UI**
<https://github.com/panaverse/styling-nextjs-projects>
3. **Todo Full-Stack App**
<https://github.com/ogzhanolguncu/min-todo>
4. **Build a Twitter Clone**
<https://www.youtube.com/watch?v=nzIsYIPCc80>

Earn While You Learn Projects

A. Build Full-Stack Next.js 13 Jamstack Templates

You will be assigned to build a template which we will sell on Theme Forest and Panaverse DAO marketplace. The Panaverse DAO will receive 20% share on the sale of the template which will be used to manage the platform. An additional 20% will be spent on marketing the template. 60% of the revenues will be distributed to the developer through the Panaverse DAO in the form of Panaverse tokens.

The Template Standard

<https://github.com/panaverse/panaverse-template-standard>

B. Build GraphQL APIs

You will be assigned to build APIs for which you will sell subscriptions on the Panaverse DAO Marketplace. The Panaverse DAO will receive 20% share on the sale of the template which will be used to manage the platform. An additional 20% will be spent on marketing the template. 60% of the revenues will be distributed to the developer through the Panaverse DAO in the form of Panaverse tokens.

Web 3 and Metaverse Specialization

Quarter III

W3-351: Developing Planet-Scale Web 3.0 Dapps

Duration: 13 Weeks

Course Description:

In this course you will learn how to develop a Web 3.0 DApps, create a project, deploy it in production, write smart contracts, unit test them, and create user interfaces for them. We will also learn to develop ERC-20 and NFT tokens, DAO's, Oracles, etc. Please note that in order to develop Web 3 applications you also need to build on top of Web 2.0 technologies which we have already covered in the previous quarters.

<https://javascript.plainenglish.io/why-you-should-learn-web-2-0-before-getting-into-web-3-0-d251625990db>

Course Outline:

1. Blockchain and Metaverse Theory

The Metaverse: And How It Will Revolutionize Everything by Matthew Ball

https://www.amazon.com/Metaverse-How-Will-Revolutionize-Everything/dp/1324092033/ref=tmm_hrd_swatch_0

Mastering Blockchain - Fourth Edition by Imran Bashir

<https://www.packtpub.com/product/mastering-blockchain-fourth-edition/9781803241067>

2. Smart Contract Development in Solidity

Solidity Programming Essentials - Second Edition By Ritesh Modi

<https://www.packtpub.com/product/solidity-programming-essentials-second-edition/9781803231181>

Solidity Learning Repo

<https://github.com/panaverse/defi-dapps-solidity-smart-contracts>

3. Dapp Development using Ethers.js and Next.js 13

Dapp Learning Repo

<https://github.com/panaverse/dapps-nextjs>

4. Tokenomics

5. Blockchain Project: Create a Token and Launch ICO/IEO/IDO

As you probably know, the ICO ("Initial Coin Offering") industry has been booming, and it's completely reinventing the way new startups kickstart themselves. In fact, have a look at Wikipedia's list of highest crowdfunding projects https://en.wikipedia.org/wiki/List_of_highest-funded_crowdfunding_projects, and you'll notice that blockchain projects absolutely dominate the list.

Understand the difference between IDO vs. IEO vs. ICO

<https://phemex.com/blogs/what-is-a-dex-ido>

Also check these links for latest listings:

ICO list at ICO Drops <https://icodrops.com>

ICO List of Best New Initial Coin Offerings <https://topicolist.com/>

Top-Rated Crypto Token Sales: List of New ICOs, STOs, IEOs and IDOs
Top-Rated Crypto Token Sales: List of New ICOs, STOs, IEOs and IDOs
<https://cryptototem.com/ico-list/>

ICO List Online <https://www.icolistingonline.com>

Binance IEO List <https://coincodex.com/ieo-list/binance/>

Binance Launchpad
<https://www.coinspeaker.com/ieo/platform/binance-launchpad/>

IEO List <https://icomarks.com/ieo>

Polkastarter <https://polkastarter.com>

Project Part 1: How to Launch a IEO on Binance Launchpad

Read How to Launch an IEO
<https://appinventiv.com/blog/how-to-launch-an-ieo/>

Your first task of the project is to make a google slides presentation on how to start a IEO on the Binance Launch Pad
<https://www.binance.com/en/support/faq/94ed108ce89d44ab8602aa3c476dfb04>

Note: Also document the alternatives to Binance Launchpad.

Project Part 2: How to Launch a IDO on Polkastarter

Review the [list of top fundraising platforms
<https://cryptorank.io/fundraising-platforms>

Your second task of the project is to make a google slides presentation on how to start a IDO on the Polkastarter <https://polkastarter.com/>

Project Part 3: Create a Pako Token

By creating a token and related contracts, you'll also learn how to handle money sent to your contracts, which should come in handy if you want to create some kind of paid decentralized service in the future.

Therefore, for the sake of this chapter, let's imagine that our Pako DApp uses its own coin – the Pako Token. We will create two contracts – one for the token itself and one for the token crowd sale (the ICO).

Now Create your own Pako ERC20 Token and deploy it on a testnet. The Token should be to use OpenZeppelin contracts. You will use the Hardhat development environment. Also, write at least twenty automated tests. We will be using Solidity and Typescript for development.

Project Part 4: Develop Crowd Sale Contract

This contract will be responsible for allowing users to exchange ETH for our Pako Token. In order to do that we need to set a price for our token (1 ETH = 100 Pako Token)

Implement a payable buyToken() function.

Emit a BuyTokens event that will log who's the buyer, the amount of ETH sent and the amount of Token bought

Transfer 75% of the Tokens to the Crowd Sale contract at deployment time. i.e. Right after the contract is deployed, we want the token contract to send 75% of our token supply to it. While 25% remain in our personal "owner" account.

Transfer the ownership of the Crowd Sale contract (at deploy time) to our frontend address so that we are able to withdraw the ETH.

You can use the openzeppelin crowd sale contracts

<https://docs.openzeppelin.com/contracts/4.x/crowdsales> however you will have to update the code to the latest solidity version.

Also write extensive tests, for example we will simply send a transaction of 1 ETH from a random account to the contract. After the transaction, we should expect the account to have received Pako, while the contract's balance should have been reduced.

Note: Before you get started writing the token contract we suggest you review access control

<https://docs.openzeppelin.com/contracts/4.x/access-control>

Project Part 5: Trying it with MetaMask

While it's always good to test your code, it's often more satisfying to see the results of your work wrapped in a nice UI. Let's see how we can deploy our contracts and get some Pako tokens into our MetaMask wallet!

We start by running our deployment scripts for the test network so that the new token contracts are uploaded and deployed.

When it's done, take note of what addresses the contracts were uploaded to and copy it!

Now head to MetaMask, and send a transaction of 1 ETH to the crowd sale contract address. If your MetaMask wallet doesn't have any ethers, remember that you can use any faucet.

After the transaction has been confirmed, you might be confused about why you can't see any tokens in your wallet. It turns out that you need to manually add the token address in MetaMask in order to "register" it – after all, there are so many tokens out there, there's no way MetaMask could list them all by default!

To do this, open the side menu and click on the "Add token" button to get started:

Once you're on the token page, click on "Add custom token" and paste in the token contract's address in the address field.

After confirming that you want to add the token, you should be able to see your Pako balance right next to your ETH balance in the wallet. How cool!

Project Part 6: Trying it with Multisignature Wallets

Read

<https://www.coindesk.com/tech/2020/11/10/multisignature-wallets-can-keep-your-coins-safer-if-you-use-them-right/>

Now use Gnosis Safe][<https://gnosis-safe.io/> with multi-sigs to do what you did in the last part.

Project Part 7: Sending Tokens using Ethers.js

Write a Typescript program to send Pako Token to some friend's address using Ethers.js.

You may follow this tutorial

<https://ethereum.org/en/developers/tutorials/send-token-ethers/>

Project Part 8 Advance: Create, Deploy, Mint, and Sell a NFT

Read this NFT tutorial series

<https://ethereum.org/en/developers/tutorials/how-to-write-and-deploy-an-nft/>

Create a NFT contract using the [OpenZeppelin ERC721 NFT Standard

<https://docs.openzeppelin.com/contracts/4.x/erc721>

You may use the Preset ERC721 contract

<https://docs.openzeppelin.com/contracts/4.x/erc721#Presets>

Deploy your NFT contract on a testnet, mint it, and view it on the MetaMask wallet and list it on OpenSea Marketplace <https://opensea.io/> for sale.

Implement a ERC721 Market

<https://ethereum.org/en/developers/tutorials/how-to-implement-an-erc721-market/>

Quarter IV

MV-361: Developing Planet-Scale Virtual and Augmented Open Metaverse Experiences

Duration: 13 Weeks

Course Description:

The Web is the Metaverse. The metaverse requires an infrastructure that connects all of the metaverses so that we can travel between them. This is only achievable with open web-based metaverses. The internet and its browsers are magical. Federated but linked experiences connect pages, people, technology, businesses, standards, and nations. It's unlike anything else. The metaverse should replicate the best qualities of the web - it should just be the web. We just have to extend online responsive design from mobile to desktop to 3D, Augmented Reality, and Virtual Reality. This course will teach you how to build the Open Social Spatial Web with WebXR and WebGPU technologies.

Course Outline:

1. Open Metaverse Web Development

Open Metaverse Learning Repo

<https://github.com/panaverse/metaverse-web>

2. Blender 3D asset Creation for the Metaverse (Remote Zoom Class)

Blender development is being funded by heavyweights in the real-time 3D (RT3D) space such as Apple, Unity, AMD, Intel, Meta, NVIDIA, AWS, Epic Games, Adobe, Microsoft, and Decentraland. It is expected to become the standard asset creation tool for metaverse.

Blender 3.3+ Download <https://www.blender.org/download/>

Blender 3.0 Beginner Tutorial

<https://www.youtube.com/playlist?list=PLjEaoINr3zgFX8ZsChQVQsuDSjEqdWMAD>

Blender 3.0 Hotkey

https://docs.google.com/document/d/1zPBgZAdftWa6WVa7UIFUqW_7EcqOYE0X743RqFuJL3o/edit

Blender Projects Textbook: Blender by Example 2nd Edition

<https://www.packtpub.com/product/blender-3d-by-example-second-edition/9781789612561>

Blender Textbook: The Complete Guide to Blender Graphics: Computer Modeling & Animation 7th Edition by John M. Blain

https://www.amazon.com/Complete-Guide-Blender-Graphics-Animation/dp/103212167X/ref=sr_1_5

Best Hardware Performance for Blender Rendering
<https://www.youtube.com/watch?v=H7T0SzdFHwg>

Assignment 1:

Build a 3D Donut using Blender 3 as shown in these video tutorials
<https://www.youtube.com/playlist?list=PLjEaoINr3zgFX8ZsChQVQsuDSjEqdWMAD>

Assignment 2:

Build a Viking Scene using Blender 3 as shown in chapter 2 of the Book [Blender by Example 2nd Edition
<https://www.packtpub.com/product/blender-3d-by-example-second-edition/9781789612561>

Assignment 3:

Modeling a Time Machine using Blender 3 as shown in chapters 3 and 4 of the Book Blender by Example 2nd Edition
<https://www.packtpub.com/product/blender-3d-by-example-second-edition/9781789612561>

Assignment 4:

Build a Modern Kitchen using Blender 3 as shown in chapters 5, 6, and 7 of the Book Blender by Example 2nd Edition
<https://www.packtpub.com/product/blender-3d-by-example-second-edition/9781789612561>

Assignment 5:

Illustrating an Alien Hero with Grease Pencil as shown in chapter 8 of the Book Blender by Example 2nd Edition
<https://www.packtpub.com/product/blender-3d-by-example-second-edition/9781789612561>

Assignment 6:

Build a 3D Sword in the Stone using Blender 3 as shown in these video tutorials
<https://www.youtube.com/watch?v=bpvh-9H8S1g>

Cloud-Native Computing Specialization

Quarter III

CN-351: Certified Kubernetes Application Developer (CKAD)

Duration: 13 Weeks

Course Description:

Cloud-native architecture and technologies are an approach to designing, constructing, and operating workloads that are built in the cloud and take full advantage of the cloud computing model. Cloud-native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

Kubernetes is an open-source system for automating deployment, scaling, and management of containerised applications. In this course you will learn how to develop cloud application using cloud-native technologies like Containers, Kubernetes, and CDK for Kubernetes.

Course Outline:

1. Kubernetes: Up and Running: Dive into the Future of Infrastructure 3rd Edition
https://www.amazon.com/Kubernetes-Running-Dive-Future-Infrastructure/dp/109811020X/ref=sr_1_3
2. Cloud Development Kit for Kubernetes
<https://cdk8s.io/>

Quarter IV

CN-361: Developing Multi-Cloud Apps using CDK for Terraform

Duration: 13 Weeks

Course Description:

Terraform lets you use the same workflow to manage multiple providers and handle cross-cloud dependencies. This simplifies management and orchestration for large-scale, multi-cloud infrastructures.

Cloud Development Kit for Terraform (CDKTF) allows you to use familiar programming languages to define and provision infrastructure. This gives you access to the entire Terraform ecosystem without learning HashiCorp Configuration Language (HCL) and lets you leverage the power of your existing toolchain for testing, dependency management, etc.

Course Outline:

CDK for Terraform

<https://developer.hashicorp.com/terraform/cdktf>

Ambient Computing and IoT Specialization

Quarter III

AC-351: Alexa Matter App Development

Duration: 13 Weeks

Course Description:

Ambient computing, also commonly referred to as ubiquitous computing, is the concept of blending computing power into our everyday lives in a way that is embedded into our surroundings - invisible but useful. In a multi-device world, people don't want to spend their life fussing with technology. An ambient approach gets the tech out of your way so you can live your life while getting the help you need. It doesn't matter what device you're using, what context you're in, whether you're talking, typing, or tapping. The technology in your life works together seamlessly.

If you were thinking that the IoT and ambient computing sound a lot alike, you aren't wrong; the two concepts are intertwined. IoT refers to the vast array of devices that connect to the internet to optimize their functionality, like smart sensors and smart speakers: ambient computing builds on that. Ambient computing focuses on the interaction between these devices once they are connected.

Matter, the next-generation smart home standard, solves many smart home pain points while bringing all our IoT devices together. Some of the biggest tech companies are working together to make Matter a unified protocol for future smart homes. These companies include Apple, Amazon, and Google. This means Apple HomeKit, Amazon Alexa, Samsung SmartThings, Google Nest, and Samsung SmartThings will support the Matter standard by default for all new devices.

In this course we will learn to build smart homes with Amazon Alexa and Matter protocol.

Course Outline:

1. **Alexa Skill Development**

<https://developer.amazon.com/en-US/alexa>

2. **Alexa with Matter Protocol**

<https://developer.amazon.com/en-US/alexa/matter>

<https://developer.amazon.com/en-US/blogs/alexa/device-makers/2022/07/alexa-live-2022-ack-sdk-matter.html>

Quarter IV

AC-361: Embedded Programming using C and Rust

Duration: 13 Weeks

Course Description:

This is an introductory course about using the C and Rust Programming Languages on "Bare Metal" embedded systems, such as Microcontrollers. We will start by introducing embedded systems and move on to learn the C++ and Rust programming languages. We'll learn about basic programming concepts using C and Rust, then we will explore key concepts in electronics, microcontrollers, and embedded programming. It is a fast-paced, thorough introduction to programming with C and Rust that will have you writing programs, solving problems, burning your code on microcontrollers, playing with GPIOs, and making things that work in no time.

Course Outline:

1. **Introduction to the Internet of Things and Embedded Systems (Weeks 1 and 2)**

<https://docs.google.com/presentation/d/14OcW4HfS2i1Db1uKOU6SrckFEFjhSLMgfnHYB3XIEZo/edit?usp=sharing>

What is the Fourth Industrial Revolution?

What is IoT?

Embedded Systems

Hardware and Software for IoT

Edge and Cloud Computing

The future of IoT is AI

Blockchain in the Internet of Things?

IoT + AI + Blockchain: The Fourth Industrial Revolution has begun

How will Matter change the IoT Infrastructure and address issues

Metaverse and IoT

Mid-Term I: Introduction to the Internet of Things (IoT) Quiz in Week 3

Total Questions: 46, Total Time: 60 minutes

2. **The C Reference Book: The C programming language**

Embedded Programming book: Internet of things with ESP8266

Download Link:

<https://drive.google.com/drive/folders/1CZFxLSFoo9rl3le7KRuhUwLEV6h4xLXN?usp=sharing>

3. **Introduction to C Part 1 (Weeks 3B, 4, and 5)**

C environment Setup for (Windows, Linux, and Mac OS systems)

Chapters 1-2 of **“The C programming language”**

- Variable names types
- Data types and sizes
- Constants
- Arithmetic operations
- Logical and relational operators
- Type conversions.
- Bitwise operators
- Conditional expressions.

Programming Assignments will also be given.

4. **C Programming Part 2 (Weeks 6 and 7)**

Topics 3 and 4 of “**The C programming language**”

- Control flow statements (else if, loops, switch, break continue)
- Function and Program structure (Returning and non-returning, scope rules, Recursion)

Programming Assignments will also be given.

5. **Introduction to Embedded systems Part 1 (Weeks 8)**

Basic Electronics and Introduction to microcontrollers

Chapter 1 of “**Internet of things with ESP8266**”

- Arduino IDE installation and env setup for ESP8266
- Burning your first code on ESP8266

Project-based Assignments will also be given.

Mid-Term II: C Programming Quiz 1 in Week 9

Total Questions: 62, Total Time: 75 minutes

6. **Introduction to Embedded systems Part 2 (Weeks 9-12)**

Chapters 2-5 of “**Internet of things with ESP8266**”

- Connecting your hardware to wifi.
- <https://randomnerdtutorials.com/wifimanager-with-esp8266-autoconnect-custom-parameter-and-manage-your-ssid-and-password/>
- Reading data from GPIOs.
- [https://randomnerdtutorials.com/esp8266-nodemcu-digital-inputs-outputs-arduino/#:~:text=ESP8266%20NodeMCU%20Read%20Digital%20Inputs,number\)%20you%20are%20referring%20to.](https://randomnerdtutorials.com/esp8266-nodemcu-digital-inputs-outputs-arduino/#:~:text=ESP8266%20NodeMCU%20Read%20Digital%20Inputs,number)%20you%20are%20referring%20to.)
- Controlling LEDs.
- Reading data from digital sensors.
- <https://randomnerdtutorials.com/esp32-dht11-dht22-temperature-humidity-sensor-arduino-ide/>
- Cloud data logging.
- <https://ttapa.github.io/ESP8266/Chap16%20-%20Data%20Logging.html>
- <https://circuitdigest.com/microcontroller-projects/nodemcu-datalogger-to-save-temperature-and-pressure-data-on-thinger-io-cloud-platform>
- <https://iotdesignpro.com/projects/nodemcu-data-logger-to-upload-data-on-webserver>

- Displaying data on the cloud.
- <https://github.com/adesolasamuel/NodeMCU-DHT-Data-to-Arduino-IoT-Cloud>
- Controlling controller remotely.
- <https://www.instructables.com/Control-ESP8266-Over-the-Internet-from-Anywhere/>
- Controlling a lamp anywhere in the world.
- Interacting with different web services.

4. Embedded Programming using Rust (Extra Weeks in the Course)

<https://www.rust-lang.org/what/embedded>
<https://crates.io/crates/esp8266>

5. Hardware Requirements:

- a. Esp8266 (Node MCU)
- b. Jumper Wires
- c. Bread Board
- d. LEDs
- e. Sensors: (Dht11, ultrasonic sensor, IR sensor)