

Robotics

Topic Name : K12

Sub-topic Name : CLASS10

Course link : <https://ineuron.ai/course/Robotics>

Course Description :-

The Introduction to Robotics Specialization covers the fundamentals of robot flight and movement, as well as how robots detect their surroundings and alter their movements to avoid obstacles, negotiate tough terrains, and complete complicated jobs like construction and disaster response. You'll learn about real-world examples of how robots have been used in disasters, how they've improved human health care, and what their future capabilities will be. The courses lead up to a capstone in which you'll learn how to programme a robot to execute various tasks like flying and gripping objects.

Course Features :-

- => Live instructor led classess
- => Completion certificate
- => Integrate academic knowledge with the tech
- => Real-time Project
- => Live Class Recording
- => Doubt Clearing
- => Assignment in all the Module
- => Quiz in every Module
- => Career Counselling
- => Completion Certificate

What you will learn :-

- => Introduction
- => Essential Tools Basics
- => Robot Car
- => Introduction to Bluetooth
- => Project: Bluetooth Robot Car

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Curriculum details :-

- => Introduction :
 - ~ *What is Robotics?*
 - ~ *Components Required*
- => Essential Tools Basics :
 - ~ *What is a breadboard?*
 - ~ *Using a Breadboard*
 - ~ *Using a Multimeter*
 - ~ *Using jumper wires*
 - ~ *Soldering Basics*
 - ~ *Servo Motor*
 - ~ *Dc motor*
 - ~ *Stepper Motor*
- => Robot Car :
 - ~ *Components of Robot car*
 - ~ *Assembly of the car*
 - ~ *Coding*
 - ~ *Testing*
- => Introduction to Bluetooth :
 - ~ *Components Using Bluetooth*
 - ~ *Bluetooth module basics*
 - ~ *Bluetooth Coding & Output*
- => Project: Bluetooth Robot Car :
 - ~ *Components of Bluetooth Car*
 - ~ *Assembly*
 - ~ *Coding*

Airflow

Topic Name : BIG DATA

Sub-topic Name : TECH STACK

Course link : <https://ineuron.ai/course/Airflow>

Course Description :-

Airflow is a tool developed by Apache for automating and scheduling tasks, data pipelines and workflows. It makes the management of data pipelines easy to manage and provides great functionalities and user interface to create pipelines.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Airflow in detail
- => Course completion certificate

What you will learn :-

- => What is Airflow?
- => Why Airflow?
- => DAGs, Schedulers, Workflows
- => HDFS
- => Capstone project

Requirements :-

- => System with minimum i3 processor or better
- => At least 4 GB of RAM
- => Working internet connection
- => Dedication to learn
- => Python
- => Docker
- => HDFS

Instructors :-

- => MD Imran :
 - ~ Working as Data Scientist with experience in solving real world business problems across different domains.

Curriculum details :-

- => Introduction :
 - ~ Welcome to course Preview
 - ~ Introduction to Apache Airflow Preview
 - ~ Conventional Scheduling Approaches
 - ~ Why move to airflow
 - ~ Basic Terminologies in Airflow DAG
 - ~ Operators
 - ~ scheduling the dags
 - ~ Executors
 - ~ Tasks and Workflow
- => Architecture of Apache Airflow :
 - ~ Architecture of Airflow
 - ~ Life Cycle of Task
- => Installation :
 - ~ Docker installation
 - ~ Airflow Installation Part 1 Preview
 - ~ Airflow Installation Part 2
- => Understanding Directories in Air :
 - ~ understanding compose files
 - ~ understanding other directories
- => Airflow UI Tour :
 - ~ First look of Airflow UI

- ~ Running Default DAG in UI
- ~ Views in UI
- ~ Understanding DAG Definition file
- ~ DAG File execution

=> What are Operators :

- ~ What are Operators

=> Project :

- ~ Project Requirements
- ~ writing project compose file part 1
- ~ writing project compose file part 2
- ~ App Password generation
- ~ writing projects dag file part 1
- ~ writing projects dag file part 2
- ~ writing projects dag file part 3
- ~ writing projects dag file part 4
- ~ creating connections in UI part 1
- ~ creating connections in UI part 2
- ~ full project explanation
- ~ mysql view table

=> Airflow CLI Basics :

- ~ Running project Dag in airflow CLI part 1
- ~ Running project Dag in airflow CLI part 2
- ~ Running project Dag in airflow CLI part 3

=> Executors in airflow :

- ~ What are executors
- ~ Sequential Executor
- ~ Local Executor
- ~ Celery Executor

Explainable AI

Topic Name : DATA SCIENCE

Sub-topic Name : MACHINE LEARNING

Course link : <https://ineuron.ai/course/Explainable-AI>

Course Description :-

Explainable AI

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => All about XAI
- => Explaining AI with Python
- => West Nile virus a case of life or death
- => Explaining Machine Learning with Facets

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Instructors :-

=> Rishav Dash :

~ This is Rishav Dash. I am a Jr. Data Scientist and mentor at INeuron.ai with working experience in computer vision, natural language processing, Machine Learning, and AOps. Hands-on experience leveraging machine learning, deep learning, transfer learning models to challenging real-world problems, and building products to solve peoples problems.

Curriculum details :-

=> Explainable AI :

- ~ Introduction to Explainable AI (XAI)
- ~ All about XAI
- ~ Explaining AI with Python
- ~ West Nile virus a case of life or death
- ~ XAI can save lives using Google Location H
- ~ Explaining Machine Learning with Facets
- ~ Microsoft Azure ML Model Interp SHAP
- ~ SHAP Implementation
- ~ Building XAI solution from scratch
- ~ AI fairness with Google_s What-if-Tool(WIT)
- ~ Local Interpretable Model-Agnostic Explanation(LIMEI)
- ~ The END

Power BI Foundations

Topic Name : DATA ANALYTICS

Sub-topic Name : POWER BI

Course link : <https://ineuron.ai/course/Power-BI-Foundations>

Course Description :-

Power BI is a luxury tool in the hands of businesses overwhelmed by the amount of data they have on hand, and we don't have any other cost-effective way to pull insights than it until now. As a result, power BI swiftly establishes itself as the world's most powerful self-service business intelligence platform and an indispensable tool for both data pros and beginners.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => Creating Reports
- => Visualization
- => Real-time insights
- => Dashboarding
- => Business intelligence workflow

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Instructors :-

- => Jayant Topnani :
 - ~ Having 2+ years teaching experience and have mentored students online & offline across all boards.

Curriculum details :-

- => Introduction :
 - ~ Difference between Business Analyst & a data analyst
 - ~ Business Understanding & Data Understanding
 - ~ Data Analysis & Data Visualization
 - ~ Data Cleaning & Preparation & methods used with some examples
 - ~ Tools for Data Analysis & Visualization
 - ~ Some charts & best practice related to them
 - ~ Some data connectors in Power BI (Excel/txt,csv)
 - ~ Basic elements of power BI their differences & use
 - ~ Ways of PBI desktop App installation & system requirements
 - ~ PBI desktop App interface Explanation
- => Data Modeling :
 - ~ Modeling Basics
 - ~ Creating Relationships
 - ~ Normalization-Denormalization
 - ~ Dimension & Fact Tables
 - ~ Relationships (Autodetect, Manual & Autodetection settings)
 - ~ Cardinality
 - ~ Active & Inactive Relationships
- => Power Query Editor :
 - ~ Some Basic Data Cleaning Operations
 - ~ Data Transformation
 - ~ Merge & Append Queries
 - ~ Interview Questions related to query Editor
- => DAX :
 - ~ DAX Basics
 - ~ Row & Filter Context
 - ~ Measures & Calculated Columns
 - ~ Some DAX Functions

=> Reporting & Dashboarding :

- ~ Various Types of Filters
- ~ Various Types of Visualizations & Formatting Options
- ~ Story telling & Dashboarding

=> Some More Connectors :

- ~ SQL server
- ~ Odata feed
- ~ Blank Query
- ~ MS Access
- ~ Real Time Data Streaming
- ~ Web
- ~ Pdf
- ~ Folder
- ~ OLE DB

=> PBI service, Mobile App :

- ~ Login PBI service
- ~ User Interface PBI service
- ~ Natural Language query & Quick Insights feature
- ~ Visual Interactions
- ~ Publishing a report to PBI service
- ~ Sharing reports with others, PBI service
- ~ How can we access reports & dashboards through PBI mobile App ?
- ~ Schedule a refresh

Class 8th Chemistry

Topic Name : K12

Sub-topic Name : CLASS8

Course link : <https://ineuron.ai/course/Class-8th-Chemistry>

Course Description :-

The Science Syllabus is elegantly designed such that it introduces the basic concepts of Science and its importance in our daily life. It will make the foundation strong for the higher classes. In this, the Chemistry section focuses on concepts like Synthetic Fibers and Plastics, Metals, Non-metals, Fossil Fuels, etc.

Course Features :-

- => Self Paced Videos
- => Completion Certificate

What you will learn :-

- => Coal and Petroleum
- => Synthetic fibers and plastics
- => Materials - Metals and Non-metals
- => Pollution of Air and Water
- => Combustion and Flame

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Instructors :-

- => Jayant Topnani :
 - ~ Having 2+ years teaching experience and have mentored students online & offline across all boards.

Curriculum details :-

- => Ch3 Synthetic Fibres & Plastics :
 - ~ Synthetic Fibres and Plastics
- => Ch4 Metals&Non-Metals :
 - ~ Material Metals and Non Metals
- => Ch5 Coal and Petroleum :
 - ~ Coal and Petroleum
- => Ch 6 Combustion and Flame :
 - ~ Combustion and Flame
- => Ch18 Pollution of Air and Water :
 - ~ Pollution of Air and Water Preview

Ethical Hacking & Penetration Testing

Topic Name : CYBER SECURITY

Sub-topic Name : CYBERSECURITY MASTERS

Course link : <https://ineuron.ai/course/Ethical-Hacking-&-Penetration-Testing>

Course Description :-

Ethical hacking is a topic that has grown increasingly essential in today's world, and it can assist individuals and companies in adopting safe IT practices and usage. This ethical hacking course will teach you those skills as well as prepare you for associated certification examinations, allowing you to demonstrate your competence.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => Introduction to the world of Security Audits and Ethical Hacking
- => Setting up your own pentesting lab
- => Getting friendly with Linux and security related commands
- => Maintaining anonymity
- => Information Gathering with Kali & Web apps
- => Introduction to web pentesting and dvwa testing environment
- => Command Injection (Low-Medium-High), Mitigation and Incident Case Study
- => Cookie manipulating and insecure session management
- => Burpsuite complete training with fundamentals
- => File upload vulnerability
- => SQL Injection
- => Boolean Injection & automation with SQL Injection

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Curriculum details :-

- => Introduction to the world of Security Audits and Ethical Hacking :
 - ~ Ethical Hacking Origin
 - ~ types of hackers
 - ~ Introduction to the world of Security Audits and Ethical Hacking
 - ~ Disclaimer and Requirements
- => Setting up your own pentesting lab :
 - ~ Grabbing tools
 - ~ Grabbing tools(CORRECTION REQUIRED)
 - ~ Installation of Virtual Box
 - ~ Installing Kali Linux
 - ~ Tour of VB & kali
 - ~ Installing Metasploitable
 - ~ Installing Windows & XP in VB
- => Getting friendly with Linux and security related commands :
 - ~ Important Network Settings & Command lines
 - ~ Basic Terminal Commands
 - ~ Managing File Permissions in Linux
 - ~ More terminal security related commands
- => Maintaining anonymity :
 - ~ Installing tor, onion routing and dark net
 - ~ Do's and Don't of Tor browser
 - ~ Proxy chaining and DNSleaktest
 - ~ Changing your Mac Address

=> Information Gathering with Kali & Web apps :

- ~ *dnsenum (information Gathering)-*
- ~ *Zone Transfer Vulnerability-*
- ~ *Information Gathering With Dig Tool-*
- ~ *Dns Tracing And Wireshark Overview-*
- ~ *Information Gathering With Websites-*
- ~ *finding Other Domains On Same Server And Comprehensive Dns Report-*
- ~ *finding Subdomains And Facebook Subdomain Brute Force Report Studying-*

=> Introduction to web pentesting and dvwa testing environment :

- ~ *intro To Web Pentesting(n)-*
- ~ *configuring Dvwa In Kali(n)-*
- ~ *dvwa Error Solving(n)-*

=> Command Injection (Low-Medium-High), Mitigation and Incident Case Study :

- ~ *What is command injection and CI low level breach*
- ~ *Breaching medium level of security*
- ~ *Command injection - High Level*
- ~ *Command injection Mitigation and secure code logic*
- ~ *Remote Code Execution incident Report*

=> Cookie manipulating and insecure session management :

- ~ *Insecure Session management & cookie manipulating flaw*

=> What is Cross Site Request Forgery? CSRF Attacks & Mitigation :

- ~ *CSRF (N)*
- ~ *CSRF()N*
- ~ *CSRF Part (N)*

=> Burpsuite complete training with fundamentals :

- ~ *HTTPS Fundamental & request response method*
- ~ *Introduction and configuring burpsuite*
- ~ *Importing HTTPS certificate in burpsuite*
- ~ *Understanding all the modules of burpsuite*

=> File upload vulnerability :

- ~ *File Upload Vulnerability (Low Security)*
- ~ *File Upload Vulnerability (Medium)*

=> File inclusion Vulnerability :

- ~ *LFI & RFI - Low Level*
- ~ *NewLFI-RFI (Medium & High level)*

=> SQL Injection :

- ~ *Sql-Lab-setup & What is database?*
- ~ *Sql Fundamentals*
- ~ *What is id and how to join the query in SQLi*
- ~ *Selecting vulnerable columns and fetching database name*
- ~ *dumping the database*

=> Boolean Injection & automation with SQL Injection :

- ~ *Boolean based queries and fundamentals*
- ~ *Boolean based live example*
- ~ *Automation with SQL Map*

=> Cross site scripting :

- ~ *XSS Reflected*
- ~ *Stored XSS & yahoo cookie stealing tale*
- ~ *DOM Based XSS & Google Lab*

=> Brute force attacks, OT Bypass & Payment Gateway Bypass :

- ~ *Brute forcing Password*
- ~ *One time password Brute force & Payment Gateway Bypass*

=> Pentesting with automated tools, report & POC making :

- ~ *Pentesting reports & POC making with Parot*
- ~ *Automation with owasp Zap*
- ~ *Httrack, WPScan & Accunetix overviews with final words*

Azure Data Engineering

Topic Name : BIG DATA

Sub-topic Name : BIG DATA ON CLOUD

Course link : <https://ineuron.ai/course/Azure-Data-Engineering>

Course Description :-

Explore how the world of data has evolved and how the advent of cloud technologies is providing new opportunities for business to explore. You will learn the various data platform technologies that are available, and how a Data Engineer can take advantage of this technology to an organization benefit.

Course Features :-

- => Self paced Recording
- => Assignment in all modules
- => Quiz in every module
- => Completion Certificate

What you will learn :-

- => Learn about the responsibilities of a data engineer
- => Find out how they relate to the jobs of other data and AI professionals.
- => Explore common data engineering practices and a high-level architecting process for a data-engineering project.

Requirements :-

- => A system with internet connection
- => Your dedication
- => Interest to learn

Instructors :-

- => MD Imran :
 - ~ Working as Data Scientist with experience in solving real world business problems across different domains.

Curriculum details :-

- => Introduction to cloud :
 - ~ Introduction to cloud computing
 - ~ cloud models
 - ~ Different cloud providers Preview
- => Regions and Availability Zones :
 - ~ Understanding Regions and Availability Zones in Azure
 - ~ creating Microsoft azure account
- => Resource Hierarchy :
 - ~ Understanding Resource Hierarchy
 - ~ Demo on Resource Hierarchy Preview
 - ~ Resource groups, subscription and management groups
- => Azure Active Directory :
 - ~ Active Directory part 1
 - ~ Active Directory part 2
- => Introduction to azure cloud computing :
 - ~ Azure services overview
 - ~ managed and unmanaged service
 - ~ demo create azure sql database service
- => Introduction to data engineer profile :
 - ~ Introduction
 - ~ data engineer role and responsibility
 - ~ introduction to data engineer technologies
- => Azure sql database :
 - ~ Module Introduction
 - ~ Introduction
 - ~ Why choosing sql server in azure
 - ~ Azure laas vs Paas database offerings
 - ~ SQL server paas deployment options
 - ~ Introduction to Azure sql server in virtual machine
 - ~ sql server in azure virtual machine
 - ~ demo part 1 sql server in azure virtual machine
 - ~ demo part 2 sql server in azure virtual machine
 - ~ introduction azure single database
 - ~ Demo Azure single database
 - ~ purchasing models and service tier
 - ~ azure database vs azure datawarehouse

- ~ *introduction elastic data pool*
- ~ *azure elastic database*
- ~ *Demo part 1 Azure Elastic database Preview*
- ~ *Demo part 2 Azure Elastic database*
- ~ *introduction managed instance database*
- ~ *azure managed instance database*
- ~ *difference between on premises and managed instance*
- ~ *service tiers for managed instance*
- ~ *management operations*
- ~ *demo managed instance*

Be A DevOps Pro Tech Neuron

Topic Name : DEVOPS

Sub-topic Name : DEVOPS MASTERS

Course link : <https://ineuron.ai/course/Be-A-DevOps-Pro-Tech-Neuron>

Course Description :-

DevOps, which is a mix of cultural principles, practices, and technologies such as Linux foundations, Docker, Kubernetes, Ansible, Terraform, ArgoCD, AWS Cloud, Git, Git, Prometheus, and others, increases an organization's capacity to build applications and services at high velocity. Become a DevOps master and increase the velocity of production now.

Course Features :-

- => Online Instructor-led learning
- => Doubt Clearing
- => Proper Roadmap for DevOps
- => Lifetime Dashboard access
- => Recording of Live Class
- => Material
- => Interview Questions
- => Resume Building
- => Career Guidance
- => Quiz in every module - Based on Real Certification Questions Based
- => Certificate
- => Industry Level Projects and Case studies
- => Major Projects
- => Weekly Assignments

What you will learn :-

- => Devops
- => Linux
- => Git
- => AWS
- => Docker
- => Ansible
- => Kubernetes
- => Terraform
- => CI CD Pipelines
- => Argo CD
- => Prometheus
- => Grafana

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Instructors :-

=> Hitesh Choudhary :

~ I like to make videos related to code and tech in my free time. I also lead a few tech teams in startups, help in hiring talent for companies. I am also on a part time traveller, with 31 countries checked off so far!

=> Saksham Choudhary :

~ Hello folks, I am AWS certified Cloud Architect Engineer. With having 5+ years of Experience in Teaching, I am currently providing cloud solutions for various products via my strong hands on DevOps Skills. I am a tech youtuber with 120k + subscriber and has taught 30,000 + students, Narcotics, Custom duty officers, Police officials and Corporate candidates.

=> Ritesh Yadav :

~ Ritesh is truly passionate about data science, machine learning and DevOps in general, he likes what he does, and is keen to learn. Currently, He is working as a Jr. Data Scientist at Ineuron.ai. He also loves to Contribute to Open Source Projects, which are mainly under CNCF Landscape. Ritesh loves to work in Cloud-Native technologies and Golang (Go). Apart from this, Ritesh has been actively involved in the open-source community for over a year, helping many open-source DevOps tools and CNCF Projects like Porter, Meshery, Keptn, TensorFlow, and Thanos through his contributions.

Curriculum details :-

=> Linux Setting up an Environment :

- ~ *Vagrant : What and Why?*
- ~ *Setting Up CentOS 7 via Vagrant in VBox*
- ~ *Basic Vagrant commands for virtual machine management*
- ~ *Vagrant Bridge Network*
- ~ *Update, Upgrade and Package Installations*

=> Understanding Linux :

- ~ *What is Linux?*
- ~ *Unix Vs Linux*
- ~ *Linux Distros & Applications*
- ~ *Cent OS vs Cent OS Stream*
- ~ *Significance of Symbol keys in Linux*

=> System Access & File System :

- ~ *Introduction to Command Prompt*
- ~ *Accessing Linux System & configuring Putty*
- ~ *Introduction to Network Related commands*
- ~ *Connect Linux VM via Putty*
- ~ *Linux File System and Structure*
- ~ *Introduction to Root?*
- ~ *Absolute and Relative Paths*
- ~ *Directory Listing Attributes*
- ~ *Playing with File & Directories*
- ~ *Copying Directories*
- ~ *Different file types in Linux*
- ~ *Find and Locate - Files and Directories*
- ~ *Changing Password*
- ~ *Wildcards (*,?,^,[],)*
- ~ *Inode, Softlink & Hard Link*

=> SysAdmin :

- ~ *Linux File Editors*
- ~ *Stream Editing Commands*
- ~ *UAM (useradd, groupadd, usermod, userdel, groupdel)*
- ~ *Enable password aging*
- ~ *Switching user with sudo module*
- ~ *User communication (Users, wall, write)*
- ~ *Active directory commands (LDAP, IDM, WinBind, OpenLDAP)*
- ~ *System Utility commands (Date, Uptime, Hostname, Uname, which, cal)*
- ~ *Jobs and Scheduling*
- ~ *Systemctl and top command*
- ~ *Kill Commands*
- ~ *Crontab*
- ~ *at commands*
- ~ *Additional cronjobs scheduling*
- ~ *Process management*
- ~ *Sys Monitoring*
- ~ *Log analysis*
- ~ *Taking control on (Shutdown, init, reboot, halt)*
- ~ *Hostname management (hostnamectl, uname, dmidecode)*
- ~ *Sys Architecture*
- ~ *Terminal controls*
- ~ *Password recovery*
- ~ *setuid & setgid*
- ~ *Environment variables*

=> Software Management :

- ~ *GNU Project*
- ~ *Compiling software from code*
- ~ *Repositories*
- ~ *Apt Cache*
- ~ *Updating & Upgrading softwares*
- ~ *Uninstalling a software*

=> Shell Scripting 1st Use case :

- ~ *How is Shell Scripting in Linux Environment?*
- ~ *Getting started with Shell Scripting*
- ~ *Naming*
- ~ *Permissions*
- ~ *Variables*
- ~ *Builtins*
- ~ *Special Variables*
- ~ *Pseudocode*
- ~ *Command substitutions*
- ~ *If Statements*
- ~ *Conditionals*
- ~ *Exit statuses*
- ~ *Return codes*
- ~ *String test conditionals*
- ~ *Use Case : Reading standard input, creating accounts, Username Conventions*

=> Shell Scripting 2nd Use case :

- ~ *Random data*
- ~ *Cryptographic Hash Functions*
- ~ *Text and String Manipulations*
- ~ *Positional Parameters*
- ~ *Arguments*

- ~ For Loops
- ~ Special Parameters
- ~ While Loops
- ~ Infinite Loop
- ~ Shifting
- ~ Sleeping
- ~ Use Case : Password Generation

=> Shell Scripting 3rd Use case :

- ~ Case Statements
- ~ Functions
- ~ Parsing command line
- ~ Parsing command line functions part 2
- ~ Finding file
- ~ Playing around with Userdel commands
- ~ Exploring archives with tar
- ~ Use Case: Deleting and Disabling User Accounts

=> Shell Scripting 4th Use case :

- ~ Cut and Awk
- ~ Cut and Awk Demonstration Script
- ~ Open Network Ports
- ~ Sort
- ~ Uniq
- ~ Use Case : Parsing log files

=> Networking :

- ~ TCP/IP
- ~ Dns
- ~ Hostnames
- ~ DHCP
- ~ Dynamic & Static addressing
- ~ Troubleshooting part 1
- ~ Troubleshooting part 2

=> Final Automation of Systems :

- ~ Configuring Mini Network
- ~ Scripting for Remote systems
- ~ Introduction to Scripting Remote commands
- ~ Scripting remote commands Advance part 1
- ~ Scripting remote commands Part 2

=> Why we need GIT :

- ~ Why GIT is important
- ~ Collaboration and Forking
- ~ Installation of GIT
- ~ Autocompletion of GIT

=> Git foundation :

- ~ GIT Architecture and Github Gitlab and bitbucket
- ~ Initializing and exploring GIT
- ~ First commit and log messages
- ~ Git checksum and SHA-1
- ~ Understanding HEAD and Checksum

=> Getting into files :

- ~ Lets do it again
- ~ Track difference between files
- ~ Delete from repos
- ~ repo reset and unstaging
- ~ Can we reset commits?

=> Git Snapshots :

- ~ checkout with previous versions
- ~ Soft, Mixed and Hard reset
- ~ Ignoring the files
- ~ What to ignore
- ~ Gitignore will not listen
- ~ This is not even in git docs

=> GIT for team managements :

- ~ Git tree listing
- ~ Git log in detail

=> GIT branches :

- ~ Git Branching basics
- ~ Creating a new branch in GIT
- ~ Checkout branches
- ~ RD of branches in GIT

=> GIT Merging :

- ~ Basics of Merging in GIT
- ~ Fast forward
- ~ Conflicts and merging in GIT
- ~ Stashing a branch
- ~ Stashing in multiple branches
- ~ Clean your stash

=> GIT and GitHub :

- ~ GIT online hosting
- ~ Creating a repo at Github
- ~ Uploading local repo to remote repo

- ~ Push for a remote collab
- ~ Merging from origin master
- ~ Assisting on open source projects

=> Tags and Tickets :

- ~ Concepts of tickets and tags
- ~ Pushing tags to github

=> Course Intro :

- ~ AWS Architect for real world

=> Getting started with AWS and IAM :

- ~ FAQ for aws architect course
- ~ Getting started with AWS and expectation
- ~ Tour of AWS console with ROOT user
- ~ AWS Infra - Region and AZ
- ~ Securing root account and MFA
- ~ Custom signin link for IAM
- ~ Why groups are created
- ~ Creating groups and users
- ~ What are roles in IAM
- ~ Temporary security credentials in IAM
- ~ Billing alarms in Cloudwatch
- ~ Password compliance
- ~ buying domain on Route 53

=> Amazon Elastic Compute Cloud -EC2 :

- ~ What is Elastic Compute
- ~ Instance types and limits
- ~ your first EC2 instance
- ~ In depth guide for EC2 options
- ~ Connecting to cloud instance
- ~ Configure an AWS web server
- ~ Stress testing, Cloud watch alarms and clean up
- ~ What are user data scripts
- ~ What is instance meta-data
- ~ Docs and hands on with Elastic IP
- ~ Custom network interface cards in AWS
- ~ creating custom AMI
- ~ Launch with custom image and clean up
- ~ Placement groups - Cluster, partition and Spread
- ~ EC2 pricing - OnDemand, spot and reserved
- ~ Just for Exam

=> Virtual Private Cloud - VPC :

- ~ Why you should focus more on VPC
- ~ Understand the default VPC
- ~ Create diagram of default VPC
- ~ CIDR deep dive
- ~ Your custom VPC
- ~ Creating subnet in custom VPC
- ~ Internet Gateway and route table
- ~ Lets complete the diagram
- ~ VPC DNS hostname and resolution
- ~ updates from corporate in VPC structure
- ~ Clean up the resources
- ~ Security groups VS NACL
- ~ Understand the next diagram for VPC
- ~ Diagram 2 - VPC and subnets
- ~ Diagram 2 - Route tables and IGW
- ~ Congigure a NAT gateway
- ~ SSH agent forwarding
- ~ Bastion host and testing of diagram
- ~ Bastion host and testing of diagram part 2
- ~ NAT instance and configurations
- ~ VPC peering connection
- ~ What are transit gateways
- ~ A use case of Endpoints in VPC
- ~ preparing logs for audit - flowLogs
- ~ Resources for hybrid cloud - VPN and more
- ~ Lets audit the logs with Athena and Glue
- ~ Egress gateway cloudhub and clean up

=> Load Balancing and scalability :

- ~ What are load balancers
- ~ Type of Load Balancer
- ~ Prep work for load balancers
- ~ Configure target groups
- ~ Creating an Application load balancer
- ~ Path and HOST based routing on domain
- ~ Cross Zone load balancer
- ~ Case of Sticky session
- ~ Clean up for ALB
- ~ Network Load Balancer
- ~ Scaling - Horizontal and Vertical
- ~ Auto Scaling Group configuration
- ~ Clean up for ASG resources

=> Route 53 in Depth :

- ~ Welcome to Route 53
- ~ What are hosted zone - Public and Private

- ~ AWS DNS records - A and Alias
- ~ Creating instance in multiple region
- ~ Route 53 Health Checks
- ~ Simple and weighted route policy
- ~ FailOver and latency based policies
- ~ Multi value and restricting content on geo location
- ~ Clean up for Route 53

=> Storage in AWS - S3 :

- ~ Lets start with AWS storage
- ~ Introduction to S3 buckets
- ~ Permissions in S3 buckets
- ~ Static website hosting in S3 buckets
- ~ S3 bucket - Versioning and encryption
- ~ S3 event notifications
- ~ Access log BILLS and requester pays
- ~ S3 storage class
- ~ Data replication - CRR and SRR
- ~ S3 Select, Athena and Redshift - Query
- ~ Data life cycle policy
- ~ Getting started with cloudfront and OAI
- ~ Setup a cloudfront and OAI for a website

=> Storage - Block and Object :

- ~ Instance Store - ephemeral
- ~ Types of EBS volume and IOPS
- ~ Creating and mounting EBS volume
- ~ Getting a snapshot of EBS
- ~ Re attach EBS volume
- ~ Data migration between AZ and Region
- ~ RAID 0 and 1 config
- ~ Creating and mounting Elastic File Storage
- ~ FSx for Windows and Lustre
- ~ Storage Gateway - Hybrid cloud
- ~ Storage Gateway NOT by LCO

=> Databases in AWS :

- ~ Introduction to Databases in AWS
- ~ OLTP vs OLAP
- ~ Production level RDS walkthrough
- ~ Create a mysql db in AWS
- ~ Multi AZ replica RDS
- ~ Creating read replicas
- ~ Read Replica VS Multi AZ deployment
- ~ AWS aurora Docs walkthrough
- ~ Getting started with DynamoDB
- ~ Creating a table in DynamoDB
- ~ Reading the DAX Docs
- ~ ElasticCache memcached
- ~ ElasticCache Redis and Redis cluster
- ~ Redshift Overview

=> Application integration in AWS :

- ~ Application integration services by AWS
- ~ Simple queue service
- ~ Creating our first queue service
- ~ FIFO vs standard queue
- ~ Delay, visibility and retention time
- ~ Dead letter queue
- ~ Long polling and short polling
- ~ Attaching lambda to SQS
- ~ Clean up all the sqs resources
- ~ Step function and simple workflow service
- ~ Amazon MQ, Rabbit MQ and other services

=> PAAS and IAAS in AWS :

- ~ Getting started with PAAS and IAAS
- ~ Cloudformation inDepth guide
- ~ Beanstack application deployment

=> Process and Migrate the Data :

- ~ Kinesis and shards
- ~ Kinesis analytics and firehose
- ~ What is Elastic MapReduce
- ~ What is Athena, Glue and Glue Studio
- ~ Import from other Virtualization Services
- ~ Database Migration service and Schema Conversion Tool

=> Security Compliance :

- ~ Security and Compliance - SOX, PCI and more
- ~ Key Management Service
- ~ Hardware Security Module in Cloud
- ~ AWS WAF and shield service
- ~ Active Directory in AWS
- ~ What is AWS Cognito
- ~ AWS single sign on
- ~ AWS Directory service

=> Container Service :

- ~ What are container service in AWS
- ~ What is Docker

- ~ *What is Elastic Container Registry*
- ~ *What are microservices*
- ~ *What is Elastic Container service*
- ~ *What is Fargate*
- ~ *What is Elastic Kubernetes Service*
- ~ *AWS walkthrough for ECS and EKS*

=> AWS Serverless :

- ~ *Getting started with AWS serverless*
- ~ *A common warning for AWS*
- ~ *Route 53*
- ~ *Get Started with S3 bucket*
- ~ *Struggle of web page hosting*
- ~ *Hosting with policies*
- ~ *GET vs POST and handling response*
- ~ *Your first lambda in AWS*
- ~ *Lambda permission and cloud watch*
- ~ *Introducing API gateway*
- ~ *Lambda for POST information*
- ~ *Post Data and CORS error*
- ~ *First look at SES*
- ~ *New user for SES and lambda*
- ~ *Sending email from SES and lambda*

=> Cracking AWS Certificate :

- ~ *How to crack AWS Certification Exams*

=> Preparing for CCP :

- ~ *How to crack AWS Certification CCP Exam*

=> Preparing for Associate/Architect Exam :

- ~ *How to crack AWS Associate/Architect Certification Exam*

=> Introduction and installation of python :

- ~ *Introduction to python course*
- ~ *Python Installation*
- ~ *Pycharm Installation on Windows*
- ~ *Installation of python on MAC*
- ~ *Installing Pycharm in MAC*
- ~ *Using VSCode for python- optional*

=> First interaction with python basics :

- ~ *Indents and comments*
- ~ *take input from user and challenge*
- ~ *input challenge solution*
- ~ *getting started with variables in python*
- ~ *numbers and strings basics in python*
- ~ *Lists and tuples basics in python*
- ~ *Dictionary in python*

=> Operations in Python :

- ~ *Arithmetic and comparison operators in python*
- ~ *Logical operations in python*
- ~ *Membership and identity operations in python*

=> Conditionals and loops :

- ~ *Introduction to conditionals*
- ~ *Design a rating system in python*
- ~ *While - Getting started with loops in python*
- ~ *First step to read documentation*
- ~ *For loop in python*
- ~ *Break keyword in python loops*
- ~ *continue and pass keywords in python*

=> Detail analysis of data types :

- ~ *Randomness in python*
- ~ *Using math library in python*
- ~ *String are powerful in python*
- ~ *Detail talk about lists in python*
- ~ *Tuples and dictionary talks in python*

=> Functions Files and Exceptions :

- ~ *getting started with functions in python*
- ~ *Multiple arguments in python*
- ~ *lambda in python*
- ~ *design custom modules in python*
- ~ *Find the day assignment in python*
- ~ *Main method and file handling in python*
- ~ *Exception handling*

=> Python challenges for fun :

- ~ *Prime number and challenges*
- ~ *range of prime numbers*
- ~ *finding factorials*
- ~ *Get matrix input and print it*

=> Object Oriented programming in python :

- ~ *Introduction to class*
- ~ *objects and constructor in python*
- ~ *Getters and setters in python*
- ~ *Inheritance from Samsung to iphone*
- ~ *Method overriding in python*

=> Database TODO App :

- ~ Read *sqlite3 documentation first*
- ~ Database helper in *sqlite3 part 1*
- ~ database helper file *part 2*
- ~ Debugging and finishing the app

=> Advance side of python :

- ~ *Iterator and generators in python*
- ~ *Maps and sets in python*
- ~ *All and any functions in python*
- ~ *Collections and deque*

=> Handling API in Python :

- ~ *Requests and JSON handling in python*
- ~ *Get a unique user every time - Project*

=> Docker Installation Basics :

- ~ *What is Docker?*
- ~ *How to install Docker and Hello World*
- ~ *What is container in Docker*
- ~ *Docker vs Virtual Machine*
- ~ *First interaction with busy box image*

=> Fundamentals of docker :

- ~ *Docker lifecycle and PS*
- ~ *Start and delete a container*
- ~ *Getting a mongodb container for fun*
- ~ *Exploring exec command*
- ~ *Multiple ways to get inside a container*

=> Custom Docker images :

- ~ *Analogy for custom docker image*
- ~ *Our first base image and custom image*
- ~ *Behind the scene for custom image*
- ~ *Creating a custom mongodb image*
- ~ *Concept of caching in docker*
- ~ *Provide a custom name for your image*

=> Project and Docker :

- ~ *Introduction to node project for docker*
- ~ *Introduction to node project for docker part 2*
- ~ *Containerize a node application*
- ~ *Performance upgrade in node project container*

=> Multi container setup :

- ~ *Introduction to multi docker container*
- ~ *A mini mongo connector project*
- ~ *Put your node code in a container*
- ~ *Introduction to docker compose*
- ~ *Connect 2 compose images in docker*
- ~ *Access the compose container app with browser*

=> Ngnix - production grade deployment :

- ~ *Ngnix A production grade docker*
- ~ *Attaching volumes in Docker*
- ~ *Types of docker files*
- ~ *Dev test and production stages*
- ~ *Understand react project for docker deployment*
- ~ *Docker for development*
- ~ *Docker for testing*
- ~ *Docker for production*

=> Docker AWS and Travis CI :

- ~ *Docker CI and AWS*
- ~ *What is CI CD Jenkins vs Travis CI*
- ~ *Moving to AWS Elastic Beanstalk*
- ~ *Moving project to github repo*
- ~ *Reading Travis CI documentation*
- ~ *Writing our 1st travis CI config file*
- ~ *AWS IAM user generation*
- ~ *Elastic Beanstalk and S3 bucket*
- ~ *Finally hosting app on AWS with CI integrated with docker*
- ~ *TURN OFF those AWS apps*

=> What is Kubernetes? :

- ~ *What is Kubernetes?*
- ~ *Introduction to Kubernetes*
- ~ *Kubernetes History*
- ~ *Kubernetes Architecture*
- ~ *Kubernetes Architecture - In-depth*

=> Provisioning Infrastructure :

- ~ *Provisioning Kubernetes Infrastructure on AWS*
- ~ *Provisioning Kubernetes Infrastructure on GCP*
- ~ *Installing Kubernetes using kubeadm*
- ~ *Setting up K8 using kubeadm*

=> Installing kubectl and minikube :

- ~ *What is minikube?*
- ~ *What is kubectl?*
- ~ *Install minikube and kubectl*

=> Installing Kubernetes Using microk8s :

- ~ *Setting up K8 using microk8's*

=> Installing Kubernetes Using K3s :

- ~ *Setting up K8's using K3's*

=> Kubernetes Components :

- ~ *Node & Pod*
- ~ *Service & Ingress*
- ~ *ConfigMap & Secret*
- ~ *Volumes*
- ~ *Deployment & StatefulSet*

=> Create and start a minikube cluster in the local environment Kubernetes CLI :

- ~ *Commands with Example (kubectl)*
- ~ *Create a pod/deployment*
- ~ *Change the pod/deployment configuration*
- ~ *Debugging pods*
- ~ *Delete pod/deployment*
- ~ *Kubernetes YAML Configuration*
- ~ *Different attributes of a Kubernetes config file*
- ~ *Creating config files*

=> Kubernetes Namespace :

- ~ *What is a Namespace?*
- ~ *4 Default Namespaces*
- ~ *Create a Namespace and resources*
- ~ *Why use Namespaces?*

=> Kubernetes Healthchecks :

- ~ *What is Ingress?*
- ~ *Creating YAML Config Files for Ingress*
- ~ *How to configure Ingress in your cluster?*
- ~ *What is Ingress Controller?*
- ~ *Demo: Configure Ingress in Minikube*
- ~ *Ingress Config based on Paths*
- ~ *Ingress Config based on Domain and Subdomain*

=> Statefulset in Kubernetes :

- ~ *What is StatefulSet?*
- ~ *Deployment of Stateful and Stateless Application*
- ~ *Deployment vs StatefulSet*
- ~ *Pod Identity*
- ~ *Scaling database applications: Master and Worker Pods*

=> Kubernetes Services :

- ~ *What is a Service?*
- ~ *ClusterIP Services*
- ~ *Headless Services*
- ~ *NodePort Services*
- ~ *LoadBalancer Services*

=> Volumes in Kubernetes :

- ~ *Persistent Volume (PV)*
- ~ *Persistent Volume Claim (PVC)*
- ~ *Storage Class (SC)*

=> Deploying Microservices App to Kubernetes Cluster :

- ~ *Microservice Overview*
- ~ *Adding Dockerfile and Dockerfile Plugins*
- ~ *Adding configurations for Service Registry*
- ~ *Creating Kubernetes Config files (YAML)*
- ~ *Implementing API Gateway*
- ~ *Deploying applications to Kubernetes Cluster*
- ~ *Scaling Application*
- ~ *Kubernetes Dashboard*
- ~ *Deleting resources from Kubernetes Cluster*

=> Ansible :

- ~ *Getting started with Ansible*
- ~ *PlayBook Run and Lab Configurations*
- ~ *Ansible Modules Yaml Syntax*
- ~ *Variables*
- ~ *Playbook Flow*
- ~ *Include And roles*
- ~ *Conditionals and Loops*

=> Terraform :

- ~ *Getting started with Terraform*
- ~ *Understand Infrastructure as Code (IaC) concepts*
- ~ *Terraform Provider Basics*
- ~ *Variables, Resource Attributes and Dependencies*
- ~ *Terraform State*
- ~ *Use the Terraform CLI*
- ~ *Read, generate, and modify configuration*
- ~ *Terraform Modules*
- ~ *Terraform Cloud*
- ~ *Intro to alternatives*

=> Pulumi :

- ~ *Getting started with pulmi*
- ~ *Syntax understanding*

~ *laac with python aws*

=> CI-CD :

- ~ *Github Actions*
- ~ *Jenkins*
- ~ *Argo CD*

=> Github workflows and Actions :

- ~ *Getting started with Github*
- ~ *Events*
- ~ *Schedulers*
- ~ *External Triggers*
- ~ *Environment Variables*
- ~ *Encrypting & Decrypting Files*
- ~ *Using Functions in Expressions*
- ~ *Strategy*
- ~ *Matrix*
- ~ *Docker containers on github actions*
- ~ *Ci Cd Workflows to automate testing and deployment*

=> Prometheus :

- ~ *Getting started with Prometheus*
- ~ *Architecture of Prometheus server*
- ~ *Installation*
- ~ *Exporters*
- ~ *PromQL*
- ~ *Client Libraries*
- ~ *Quantification of Instruments*
- ~ *Recording Rules*
- ~ *Alerting*
- ~ *Create Routing Tree for alerts*
- ~ *PagerDuty - Slack Alerts*
- ~ *BlackBox Exporters*
- ~ *Pushgateway*
- ~ *Service Gateway*
- ~ *Aws With Prometheus*
- ~ *Prometheus Http API*

=> Grafana :

- ~ *Introduction , Setup and Configuration*
- ~ *Grafana UI Tour*
- ~ *Integration with different data sources*
- ~ *Grafana Templates*
- ~ *Grafana Dashboards Introduction*
- ~ *Application Dashboards*
- ~ *Managing Dashboards*
- ~ *Dynamic DashBoards*
- ~ *Security and Administration of Grafana*

DSA with Java

Topic Name : DATA STRUCTURE

Sub-topic Name : DSA WITH JAVA

Course link : <https://ineuron.ai/course/DSA-with-Java>

Course Description :-

This course has been designed to help you become a complete and professional Java developer at the conclusion of the course, rather than only teaching essential Java skills. After completing this course, you will have a thorough understanding of various Data Structures and Algorithms in Java which will further enhance your career as a java developer.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => Problem Solving
- => Time-based DSA
- => Big O notation
- => Time and space complexity
- => Recursion
- => Power program theory
- => Combination theory
- => Stacks, queues, linked lists, trees
- => Searching, sorting, hashing

Requirements :-

- => System with minimum i3 processor or better
- => At least 4 GB of RAM
- => Working internet connection
- => Dedication to learn

Instructors :-

=> Hitesh Choudhary :

~ I like to make videos related to code and tech in my free time. I also lead a few tech teams in startups, help in hiring talent for companies. I am also on a part time traveller, with 31 countries checked off so far!

Curriculum details :-

- => Introduction to DSA :
 - ~ Why we need Data structures and algorithms Preview
 - ~ Time based approach Preview
 - ~ Concept of Big O and graphs Preview
 - ~ Data Structures and Algorithms HB
- => Problem Solving :
 - ~ Start with a challenge - reverse string
 - ~ Reverse a string - solution
 - ~ Interview approach to solve a problem
 - ~ Classic interview steps for DSA problems
- => Data Structure Introduction :
 - ~ Memory process - Stack and Heap
 - ~ Physical and logical data structures
 - ~ Abstract Data Types - ADT Preview
- => Recursion in depth :
 - ~ Introduction to recursion
 - ~ Tracing the recursion tree
 - ~ Trace tree assignment
 - ~ Trace tree solution
 - ~ Types of Recursion

- ~ Complex recursion tree
- ~ What is Factorial
- ~ DSA08 Factorial program in JAVA
- ~ Fibonacci series THEORY
- ~ Fibonacci series and its version JAVA Code
- ~ What is Power Program
- ~ Power Program JAVA code
- ~ What is a Combination Program
- ~ Combination Program JAVA code
- ~ Classic Tower of Hanoi problem
- ~ Classic Tower of Hanoi JAVA code

=> Linked List in depth :

- ~ Introduction to Linked List Preview
- ~ Add value in linked list - cases
- ~ Push Append and insert in LinkedList - JAVA code
- ~ Deletion of linked list THEORY.
- ~ Deletion in linked list JAVA code
- ~ Delete complete linked list JAVA code
- ~ Count all nodes in linkedlist JAVA code
- ~ Reversing a linked list THEORY
- ~ Reversing a linked list JAVA code

=> Circular Linked List in Depth :

- ~ Circular linked list THEORY Preview
- ~ Circular Linked List push JAVA code
- ~ Traverse a circular linked list JAVA code
- ~ Deletion in circular linked list JAVA code
- ~ count nodes in circular linked list JAVA code
- ~ convert linked list to circular linked list JAVA code

=> Doubly Linked List in Depth :

- ~ Theory for doubly linked list Preview
- ~ Doubly linked list push JAVA code
- ~ Insert After in doubly linked list JAVA code
- ~ add to last in doubly linked list JAVA code
- ~ Traverse a doubly linked list JAVA code
- ~ Deleting a node in doubly linked list JAVA code

=> Stack and Queue :

- ~ Stack - Push and Pop operation THEORY
- ~ Stack operations with JAVA code
- ~ Queue concept THEORY Preview
- ~ Queue implementation in JAVA code
- ~ Circular queue THEORY
- ~ Circular queue JAVA code

=> Binary Search Tree :

- ~ What is Binary Search tree and creation THEORY update
- ~ Insertion and Deletion in BST THEORY
- ~ InOrder Traversal of BST THEORY
- ~ Pre Order traversal in BST THEORY
- ~ Post order traversal in BST THEORY
- ~ Creating a Binary Search tree JAVA code
- ~ search a key in BST JAVA code
- ~ Insertion in BST JAVA code
- ~ deletion of key in BST JAVA code
- ~ inorder preorder and postorder traversal in BST JAVA code

=> Hashing :

- ~ What is Hashing THEORY
- ~ Hash chaining with linked list
- ~ Linear Hash Shifting
- ~ Square hash shifting

=> AVL Tree :

- ~ What is AVL tree and height
- ~ Finding balance factor
- ~ Left Left and Right Right Rotation in AVL Tree
- ~ LR and RL rotation with 1 trick
- ~ Creating a AVL tree - Important
- ~ Deletion in AVL Tree.

=> HEAP :

- ~ Heap - Max and min Heap
- ~ Insertion and deletion in HEAP

=> Sorting algorithms :

- ~ Categories of sorts
- ~ Selection sort - Theory
- ~ Selection sort - Java Code
- ~ Bubble Sort - Theory
- ~ Bubble Sort - Java Code
- ~ Insertion sort - Theory
- ~ Insertion sort - Java Code
- ~ Quick Sort - Theory
- ~ Quick Sort - Theory part 2
- ~ Counting Sort - Theory
- ~ Merge Sort Theory
- ~ Merge sort JAVA code
- ~ Counting Sort - Java Code

Vedic Math

Topic Name : K12

Sub-topic Name : CLASS8

Course link : <https://ineuron.ai/course/Vedic-Math>

Course Description :-

This course will help you solve complex mathematical problems using Vedic Mathematics. This course is curated on a set of concepts that will help you improve your calculations to an extent where before you pick up a pen, you would find the answers by simplifying calculations into simple steps. If you despise numbers, This course will help you interactively appreciate the beauty of mathematics.

Course Features :-

- => Online Instructor-led learning
- => Practical Implementation
- => Integrate academic knowledge with the tech
- => Real-time Project
- => Live Class Recording
- => Doubt Clearing
- => Assignment in all the Module
- => Quiz in every Module
- => Career Counselling
- => Completion Certificate

What you will learn :-

- => Introduction about Vedic math's
- => Benefits of Vedic math's
- => Addition of numbers
- => Subtraction of numbers
- => Multiplication of numbers
- => Division of numbers
- => Square of a number
- => Cube of a number
- => Square root of a number
- => Cube root of a number

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Curriculum details :-

- => Introduction to VEDIC MATHS :
 - ~ What is VEDIC MATHS?
 - ~ Benefits of Vedic Maths
 - ~ Why we learn VEDIC MATHS ?
- => Basics VEDIC MATHS :
 - ~ Find the complement of 1 digit number/ 2 digit number/ 3 digit number/ any digit number?
 - ~ Tables for 9/19/29/39/129/149/.
 - ~ All from 9 and the Last From 10
 - ~ Multiplication of 2 numbers where number of digits are same in both number and sum of unit digit number is 10
 - ~ Multiplication of 2 numbers where digits are same in both numbers except unit digit number
 - ~ Multiplication with 11
 - ~ Multiplication with 12
 - ~ Universal multiplication like multiplication for following exs:- 1) $2^2 \times 2$ 2) $3^3 \times 3$ 3) $4^4 \times 4$ 4) $2^3 \times 5$ 5) $3^4 \times 6$ 6) $2^4 \times 7$ 7) $3^2 \times 2$ etc (called as URDHVA TIRYA GAMYAM)
 - ~ Square of different types of number (For example, whose unit digit /the last digit is 1/5/6/4
 - ~ Cubes of 2 digit number
- => Intermediate :
 - ~ Division of any number by 5, 8 & 98
 - ~ Division of any number by 11
 - ~ Division of any number by 12-19
 - ~ Division of any number by 25, 50 and 100

- ~ Division by factors
- ~ Percentages
- ~ Addition of odd, even series of numbers
- ~ Multiplication of numbers ending with 5
- ~ Multiplication of numbers with 15
- ~ Multiplication of whole number with mixed fractions

=> Advanced VEDIC MATHS :

- ~ Division of whole number with mixed fractions
- ~ Addition of special fractions
- ~ Square of any number nearer to base
- ~ Multiplications with 9/99/999 when
- 1) Multipliers are same digit
- 2) Less digits than multiplier
- 3) More digits than multiplier
- ~ Different types of base multiplication for ex. $97 * 94$ (less than base 100), $14 * 12$
- ~ Multiplication of numbers like $(42 * 46)$ considering primary & secondary base
- ~ different (in this ex, 10 & 50 are two different base)
- ~ Cubes of numbers closer to bases
- ~ Division (Nikhilam method where divider is less than base number or nearer to base number)
- ~ Quickest division by 9,99,999,9999
- ~ Repeated digit base number squares
- ~ Vinculum of number at Unit and tens places
- ~ Squares by duplex method
- ~ Square root
- ~ Cube root

=> Application: :

~ We will make a UI where we will provide a set of questions where kids can answer those questions & check if it is correct or not/ Similarly, In the backend, they can do logic according to Vedic formula.
Plus, we will provide a hint that explains the Vedic formula/procedure to find the solution in the front end.

Chatbot Using Google Dialogflow

Topic Name : DATA SCIENCE

Sub-topic Name : CHATBOT

Course link : <https://ineuron.ai/course/Chatbot-Using-Google-Dialogflow>

Course Description :-

Google Dialog flow Chatbot

Course Features :-

=> Lifetime Dashboard

=> Free Course

What you will learn :-

=> End to End Chatbot with Deployment And Custom Service Integration

=> Chatbot Fundamentals

=> Understanding The Framework

=> Business Problem- Let's Build The Chatbot

=> Building The Solution

=> Implementing The Fulfillment

=> Integration With Telegram

=> Summary And Further Work

Requirements :-

=> Programming Understanding

=> NLP understanding

Curriculum details :-

=> End to End Chatbot with Deployment And Custom Service Integration

=> Chatbot With Google Dialogflow- Chatbot Fundamentals :

~ Introduction to chatbot Preview

=> Chatbot With Google Dialogflow- Understanding The Framework

=> Chatbot With Google Dialogflow- Business Problem- Let's Build The Chatbot

=> Chatbot With Google Dialogflow- Building The Solution

=> Chatbot With Google Dialogflow- Implementing The Fulfillment

=> Chatbot With Google Dialogflow- Integration With Telegram

=> Chatbot With Google Dialogflow- Summary And Further Work

R Programming Job Preparation

Topic Name : PROGRAMMING

Sub-topic Name : R

Course link : <https://ineuron.ai/course/R-Programming-Job-Preparation>

Course Description :-

R had over 2 million users globally in 2012, according to Oracle, reinforcing R's position as the premier programming language in statistics and data research. Every year, the number of R users increases by roughly 40%, and a rising number of businesses are incorporating it into their daily operations. This course is designed for individuals who are willing to start their journey with R programming to get into the field of Data Science or Machine Learning.

Course Features :-

- => Roadmap
- => Resume preparation
- => Interview Questions
- => Quizzes
- => Assignments
- => Downloadable resources
- => Completion certificate

What you will learn :-

- => Statistics overview
- => In depth discussion for various types of questions asked in interviews and how to approach them
- => Discussion on resume building.

Requirements :-

- => Prior Understanding of R Programming.
- => A system with a decent internet connection
- => Dedication

Instructors :-

- => Shlok Pandey :
 - ~ Content Developer (Deep Learning)

Curriculum details :-

- => Course Introduction :
 - ~ Explain What is R? Preview
 - ~ What is GUI in R ?
 - ~ What is CLI in R ?
 - ~ What is Data Mining & what Data Miners do in R?
 - ~ Who & when was R discovered?
 - ~ Why should you adopt R Programming language ?
- => Introduction to basics of R :
 - ~ What are Programming Features of R? Preview
 - ~ What are the applications of R?
 - ~ Compare R with Other technologies.
 - ~ Why R is Important ?
 - ~ Is R a slow language?
 - ~ Explain main features to write R code that runs faster?
 - ~ What is SAS and SPSS in R?
 - ~ Why is R important for data science?
 - ~ Why is R Good for business?
 - ~ What is Visualization in R?
 - ~ What are R topical programming and statistical relevance?
 - ~ What are the statistical and programming features of R?
 - ~ What are the advantages of R?
 - ~ What are the disadvantages of R?
 - ~ Why R language?
- => R Analysis :
 - ~ What is Predictive Analysis in R?
 - ~ What is the Predictive analysis process in R?
 - ~ What is the need for Predictive Analysis in R?
 - ~ What is Descriptive analysis in R?
 - ~ What are Descriptive analysis methods in R?
- => R Datatypes :
 - ~ What is R studio and how to use it?

- ~ What are R data types?
- ~ How many types of data types are provided by R?

=> R Vector :

- ~ What is the main difference between an Array and a matrix?
- ~ What is R vector?
- ~ How many types of vectors are present in R?
- ~ What is an Atomic vector and how many types of atomic vectors are present in R?
- ~ What is recycling of elements in an R vector? Give an example.

=> R Lists :

- ~ What is R lists?
- ~ Explain how to create a list in R?
- ~ Explain how to access list elements in R?
- ~ Explain how to manipulate list elements in R?
- ~ Explain how to generate lists in R?
- ~ Explain how to operate on lists in R?
- ~ Can we update and delete any of the elements in a list?
- ~ How many types of object are present in R?

=> R Functions :

- ~ What are R Functions?
- ~ What are features of R functions?
- ~ What is function definition?
- ~ What are the components of R functions?

SQL Projects

Topic Name : DATA ANALYTICS

Sub-topic Name : SQL

Course link : <https://ineuron.ai/course/SQL-Projects>

Course Description :-

SQL is utilised for a wide range of things, including banking, music, social media, data analysis, and so on. The majority of firms rely on huge, relational databases and are continually on the lookout for SQL experts. This course covers real world scenario based projects to gain hands-on knowledge and implement in real time to build business solutions.

Course Features :-

- => Project source codes
- => Quizzes
- => Assignments
- => Downloadable resources
- => Completion certificate

What you will learn :-

- => Start with Entity Relationship Model(ERM) logic.
- => Build stored procedure based on the business use case.
- => Database schema design

Requirements :-

- => Prior knowledge of SQL.
- => A system with internet connection.
- => Your dedication

Instructors :-

- => MD Imran :
 - ~ Working as Data Scientist with experience in solving real world business problems across different domains.

Curriculum details :-

- => Python database connectivity (MYSQL) :
 - ~ Installing mysql Preview
 - ~ Database connectivity part 1
 - ~ Database connectivity part 2
 - ~ Database connectivity part 3
 - ~ Database connectivity part 4
- => Bank management system :
 - ~ Bank management system flow chart Preview
 - ~ Bank management system part 2
 - ~ Bank management system part 3
 - ~ Bank management system part 4

Manual Testing Foundations

Topic Name : TESTING

Sub-topic Name : MANUAL TESTING

Course link : <https://ineuron.ai/course/Manual-Testing-Foundations>

Course Description :-

This course will help you get started with Software Testing. We will discuss different terms and terminologies to develop a QA mindset. We will also discuss about roles and responsibilities of a software tester and what are the day-to-day activities that you have to perform as a tester. You will also learn STLC (Software Testing Life Cycle) and its different phases and at the end, we will use a very useful in-demand tool called "JIRA".

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => Objectives of Testing
- => Scope of Testing
- => Prerequisites of becoming a Tester
- => Potential growth in the Software Testing Career
- => Roadmap to a testing career
- => Phases of Testing
- => Unit Testing
- => Integration Testing
- => System Testing
- => UAT Testing -Alpha & Beta Testing
- => Deployment Process

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Instructors :-

=> Kiran Sahu :

~ QA Manager with 12+ years of professional experience, worked in Brands like Infosys, Delhivery, Mydala, Aurea, Jive, Crossover, Agama Solutions & OSTC, have experience of working in global platforms and with multinational professionals. Strong domain knowledge on Retail, Logistics, Banking, Trading, Ecommerce Applications. Experience in Training and Mentoring Candidates all across the globe on Software Testing, MySQL and Agile.

Curriculum details :-

=> Getting Started with Software Testing and STLC :

- ~ Basics of Software Testing
- ~ Need of Software Testing
- ~ Objectives of Testing
- ~ Scope of Testing
- ~ Prerequisites of becoming a Tester
- ~ Potential growth in the Software Testing Career
- ~ Roadmap to a testing career
- ~ SDLC
- ~ STLC
- ~ SDLC Vs STLC

=> Phases of Testing and Types of Testing :

- ~ Phases of Testing
- ~ Unit Testing
- ~ Integration Testing
- ~ System Testing
- ~ UAT Testing -Alpha & Beta Testing

- ~ *Deployment Process*
- ~ *Ecommerce Project Example*
- ~ *Types of Testing*

=> JIRA- A Complete Overview :

- ~ *Introduction to JIRA*
- ~ *Introduction to Agile*
- ~ *What is Scrum*
- ~ *What is Sprint Cycles*
- ~ *Importance of Jira in Agile*
- ~ *Project Management using Jira in Agile*
- ~ *Bug Tracking using Jira in Agile*
- ~ *Sprint Report- Burndown Chart in Jira*

IOT

Topic Name : IOT

Sub-topic Name : IOT MASTERS

Course link : <https://ineuron.ai/course/IOT>

Course Description :-

This course will help you to grab the fundamentals of IOT.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => IoT Architecture & Communication Protocols
- => MQTT Introduction
- => MQTT Installation
- => MQTT Pub Sub Test
- => MQTT ACL
- => MQTT ACL username creation
- => MQTT ACL python program
- => RPi4 Sensor Data through MQTT
- => MongoDB Introduction
- => Introduction to Azure IoT Hub
- => Introduction to Azure Storage Account

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Curriculum details :-

- => Course Introduction :
 - ~ Course Introduction
- => IoT Architecture :
 - ~ IoT Architecture & Communication Protocols
- => Introduction to MQTT :
 - ~ MQTT Introduction
 - ~ MQTT Installation
 - ~ MQTT Pub Sub Test
 - ~ MQTT Debugging methods
 - ~ MQTT QoS
 - ~ MQTT sending json file
 - ~ MQTT listener configuration
 - ~ MQTT Security Methods
 - ~ MQTT username password
 - ~ MQTT Client Tools
- => MQTT with Python :
 - ~ MQTT Python Software Development
 - ~ Jupyter notebook hello world
 - ~ Python mqtt libraries
 - ~ Paho mqtt installation
 - ~ Paho mqtt sample test
 - ~ Python simulation data
 - ~ Simulation data program
 - ~ Publish simulation data
 - ~ Subscribe simulation data
 - ~ Username password in python program
 - ~ MQTT ACL
 - ~ MQTT ACL username creation

~ MQTT ACL python program

=> RPi4 Sensor Data through MQTT :

~ Raspberry Pi 4 Introduction

~ Rpi python program for mqtt publish

~ Rpi sensor interfacing

~ Replacement of simulation with actual sensor data

=> MongoDB Introduction :

~ NoSQL databases

~ NoSQL MongoDB Properties

~ MongoDB Installation

~ MongoDB CLI

~ MongoDB Python Library pymongo

=> IoT with Cloud Computing :

~ Cloud Computing and IoT Services

=> Introduction to Azure IoT Hub :

~ Azure IoT Hub Introduction

~ Azure IoT Hub In Depth

~ Your First Azure IoT Device

~ Your First Azure IoT Device Continued

~ D2C Azure Developer Options

~ D2C Python Code

~ Tools for monitoring messages

~ C2D Python code

~ Azure IoT Hub Python SDK

=> Introduction to Azure Storage Account :

~ Azure Storage Account Service Introduction

~ Create blob storage account

~ Saving d2c data in container blob

=> Introduction to X509 Certificates :

~ Introduction to X509 certificates

~ Openssl for working with X509

~ Generating self signed certificate using openssl

~ D2C using Self Signed certificates

Solidity Live Class

Topic Name : BLOCKCHAIN

Sub-topic Name : SOLIDITY

Course link : <https://ineuron.ai/course/Solidity-Live-Class>

Course Description :-

Solidity & Solana Blockchain course is designed to provide an in depth knowledge on various aspects & concepts of blockchain & Solidity. A step by step learning will be help to focus on each & every parameter of Blockchain. This course will take you into a deep dive into the state of the art blockchain technology and how to go about writing smart contracts in the ethereal platform. Moreover, this is a project-ready course which will help you take whatever you learn and apply it into a real-world portfolio-ready app, which you can showcase to the world.

Course Features :-

- => Onine Live Classes
- => Doubt Clearing
- => Live-Class Recording
- => Real-time Project
- => Assignment in all modules
- => Quiz in every module
- => Career Counselling
- => Completion Certificate

What you will learn :-

- => Solidity Fundamentals
- => Smart Contracts in Solidity
- => Smart Contract Best Practices
- => What are ICO and what are tokens
- => Understanding about ERC-20
- => Solana Blockchain
- => Web 3.0
- => Intro to IPFS
- => Oracles
- => DeFi
- => NFTs

Requirements :-

- => Knowledge of Web Designing will be Advantageous.

Instructors :-

=> Sanjeevan Thorat :

~ Data Scientist and Blockchain developer, with experience in developing and managing end to end solutions. I have hands-on experience in Python Programming Language, Machine Learning Deep Learning and Natural language processing. Blockchain development experience in smart contracts, Decentralised Finance applications, DAOs, NFTs and Oracles running on Ethereum and Polygon blockchains. I have worked with various clients for different industry projects in the blockchain space. I specialize in building smart contracts on the Ethereum blockchain along with JavaScript integration for enhancing user experience to generate maximum returns on investment.

Curriculum details :-

- => Introduction :
 - ~ Introduction to course
- => Solidity Fundamentals :
 - ~ Smart Contracts in Solidity
 - ~ Basic-Smart-Contract-Part1
 - ~ Basic Smart Contract Part 2
 - ~ Data types and Variables - part 1
 - ~ Data types and Variables - part 2
 - ~ Functions
 - ~ Storage vs Memory
 - ~ Events and logs
 - ~ Factory contract
 - ~ Security Of Smart Contracts
 - ~ Inheritance
 - ~ Inline Assembly

~ *Application Binary Interface*

=> Smart Contracts Pitfalls, Testing and Debugging :

~ *Unit tests*

~ *Integration Tests*

~ *Javascript tests*

~ *Smart Contract Best Practices*

=> Creating our own cryptocurrency on Ethereum Network :

~ *What are ICO and what are tokens*

~ *Understanding about ERC-20*

~ *Writing code for our cryptocurrency*

~ *Safe Math*

~ *Creating the cryptocurrency*

~ *Deploying it to the network*

=> Solana Blockchain :

~ *Introduction to solana Blockchain*

~ *Creating our own cryptocurrency on the Solana Network using CLI - part 1*

~ *Creating our own cryptocurrency on the Solana Network using CLI - part 2*

~ *Creating our own cryptocurrency on the Solana Network using CLI - part 3*

~ *Creating our own cryptocurrency on the Solana Network using Javascript - part 1*

~ *Creating our own cryptocurrency on the Solana Network using Javascript - part 2*

~ *Creating our own cryptocurrency on the Solana Network using Javascript - part 3*

~ *Creating our own cryptocurrency on the Solana Network using Javascript - part 4*

=> Web 3.0 & Connecting everything into a project :

~ *What is Web 3.0 ?*

~ *iNeuron Marketplace - part1*

~ *iNeuron Marketplace - part2*

~ *iNeuron Marketplace - part3*

~ *iNeuron Marketplace - part4*

~ *iNeuron Marketplace - part5*

~ *iNeuron Marketplace - part6*

~ *iNeuron Marketplace - part7*

~ *iNeuron Marketplace - part8*

=> A little more about ethereum :

~ *Ethereum naming service*

~ *Intro to IPFS*

~ *Oracles*

~ *DeFi*

=> NFTs :

~ *What are NFTs and ERC721*

~ *Create Your own NFT part 1*

~ *Create Your own NFT part 2*

~ *Create Your own NFT part 3*

~ *Create Your own NFT part 4*

Django

Topic Name : WEB DEVELOPEMENT

Sub-topic Name : DJANGO

Course link : <https://ineuron.ai/course/Django>

Course Description :-

Django is a Python-based web framework. You'll be developing and learning about Django one step at a time in this course. We'll go through all you need to know about using Python, Django and other web technologies to create a website.

Course Features :-

- => Quizzes
- => Assignments
- => Hands-on practicals
- => Downloadable resources
- => Completion certificate

What you will learn :-

- => Django overview
- => Project implementation
- => Project deployment

Requirements :-

- => Basic knowledge of Python programming
- => A system with stable internet connection
- => Your dedication

Instructors :-

=> Khushali Shah :

~ A data scientist having rich experience working with MNCs and start-ups in the field of data science and machine learning. She has expertise in Chatbot development for various domains & been developing professionally for 6+ years with diverse job history. She also had positions in software module development, web app development, functional designs, requirement gathering, client interaction, and server setup/admin & can help everywhere in the stack; she loves wearing multiple hats to an extent. She also believes in enhancing her skills by training and learning new things day by day.

Curriculum details :-

=> Introduction :

~ Overview Preview

=> Django core :

- ~ Django features Preview
- ~ Creating a project
- ~ The development server
- ~ Creating the Polls app
- ~ Write your first view
- ~ `path()` argument: route
- ~ `path()` argument: view
- ~ `path()` argument: name

Full Stack Blockchain Development Tech Neuron

Topic Name : BLOCKCHAIN

Sub-topic Name : BLOCKCHAIN MASTERS

Course link : <https://ineuron.ai/course/Full-Stack-Blockchain-Development-Tech-Neuron>

Course Description :-

Full Stack Blockchain Development course is a live mentor-led certification program with by iNeuron. In this course you will learn the entire stack required to work in Permissionless Blockchain development. This course focuses on latest Blockchain industry standards like Ethereum Blockchain, Solidity, Decentralized Autonomous Organisations, Decentralized Finance, Non Fungible Tokens, Polygon Network, Polkadot Blockchain, Oracles along with complete development stack in Javascript and many more Blockchain concepts.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => Web 1.0 vs Web 2.0 vs Web 3.0
- => What is Blockchain technology?
- => Bitcoin Blockchain
- => Ethereum Blockchain
- => Solidity
- => Oracles
- => DAO
- => DeFi
- => NFT
- => Layer 2 Blockchain
- => Truffle Suite
- => Hardhat
- => Polkadot

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Curriculum details :-

- => Course Introduction :
 - ~ Course overview
 - ~ A brief history of internet technologies
 - ~ Web 1.0 vs Web 2.0 vs Web 3.0
 - ~ What is Blockchain technology?
 - ~ Why do we need Blockchain technology?
 - ~ The connected world and the Blockchain: A disruptive computing paradigm
 - ~ Centralized vs Decentralized networks
 - ~ Distributed Systems overview
- => Web Development :
 - ~ What is Web Development?
 - ~ Client-Server Architecture
 - ~ What are APIs?
 - ~ What is Front-end web development?
 - ~ What is Back-end web development?
 - ~ Components of Full-Stack Web Development Applications
- => HTML :
 - ~ How do websites work?
 - ~ Preview
 - ~ HTML vs CSS vs Javascript

- ~ HTML files
- ~ Doctype & HTML Boilerplate
- ~ Spaces & Line Breaks
- ~ Heading Tag
- ~ Paragraph & Pre Tag
- ~ Difference between Elements, Attributes & Tags
- ~ Comments
- ~ Useful Tags
- ~ Nesting of Tags
- ~ Extensions in HTML
- ~ Live Server in VSCode
- ~ Formatting Tags
- ~ Article in HTML
- ~ Time & Address Tag
- ~ Quote & Cite
- ~ Strike
- ~ Progress Bar
- ~ Anchor Tag Styling
- ~ Image Tag
- ~ HTML Table
- ~ List
- ~ Input Tags,iframe
- ~ Forms
- ~ Video & Audio
- ~ iframe
- ~ Embed pdf
- ~ Maps
- ~ Symbols
- ~ Meta Tags
- ~ SVG
- ~ Emoji

=> CSS :

- ~ CSS Introduction
- ~ Inline vs Internal vs External
- ~ Priority between Inline, Internal & External
- ~ Multiple Properties in Single Element
- ~ Types of Selectors
- ~ Priority between Id, Class & Element
- ~ Comments
- ~ Colors
- ~ Background
- ~ Border
- ~ Height & Width
- ~ Padding
- ~ Margin
- ~ Box Model
- ~ Text Properties
- ~ Anchor Tag Styling
- ~ Fonts
- ~ Cursor
- ~ !Important in CSS
- ~ Box Shadow
- ~ Opacity
- ~ Filter
- ~ Gradient
- ~ Overflow
- ~ List
- ~ Tables
- ~ Box Sizing
- ~ Inherit & Initial
- ~ Object Fit
- ~ Pseudo Classes
- ~ Pseudo Elements
- ~ Display
- ~ Position
- ~ Z-Index
- ~ Floats
- ~ 2D Transform
- ~ Transitions
- ~ Flex
- ~ Flex Direction & Wrap
- ~ Justify & Align in Flex
- ~ Order in Flex
- ~ Grow & Basis in Flex
- ~ Align Items in Flex
- ~ Grids
- ~ Rows, Columns & Gap in Grids
- ~ Justify & Align in Grids
- ~ CSS Validator (Final Video)

=> Javascript :

- ~ Introduction
- ~ Running Javascript in Browser
- ~ Console
- ~ Strings & Numbers
- ~ var, let & const
- ~ Data Types
- ~ Type Conversions

- ~ Arithmetic Operators
- ~ Assignment Operator
- ~ Comparison Operator
- ~ Logical Not, Or and And
- ~ Swap Numbers
- ~ String Handling
- ~ String Searching
- ~ Arrays
- ~ Objects
- ~ Dates
- ~ Maths
- ~ If & Else
- ~ Challenge - If & Else
- ~ Switch Case
- ~ Challenge - Switch Case
- ~ JS Loops
- ~ For Loops
- ~ Nested Loops
- ~ Break & Continue
- ~ Arrays, Strings & Objects
- ~ For-in
- ~ For-of
- ~ While Loops
- ~ Do while Loops
- ~ Loops Exercises
- ~ Functions
- ~ Variable Scopes in Functions
- ~ Nested Functions
- ~ Parameters & Arguments
- ~ How function is useful?
- ~ Return in Function
- ~ Anonymous Functions
- ~ Calculator Exercise
- ~ Arrow Functions
- ~ forEach
- ~ maps
- ~ String Literals
- ~ Filter, Reduce & Every
- ~ Spread Operator
- ~ Challenge
- ~ Window & Document
- ~ Document Access
- ~ innerText & innerHTML
- ~ HTML Calculator
- ~ Query Selector
- ~ Styling in JS
- ~ Advance DOM Manipulation
- ~ Events
- ~ Basic Events
- ~ Time Events
- ~ Pop-up Boxes
- ~ Error Handling
- ~ Form Validation
- ~ Asynchronous JS
- ~ this keyword
- ~ useStrict
- ~ Hoisting
- ~ Local Storage
- ~ Session Storage
- ~ Cookies
- ~ Cookies vs Local Storage vs Session Storage
- ~ JSON vs Object literals
- ~ API
- ~ Fetching
- ~ Methods & Status Codes
- ~ Post Method
- ~ Put Method
- ~ Guess the Number
- ~ Generators
- ~ Regex

=> The JavaScript Standard Library :

- ~ The JavaScript Standard
- ~ Sets and Maps
- ~ Typed Arrays and Binary Data
- ~ Pattern Matching with Regular Expressions
- ~ Dates and Times
- ~ Error Classes
- ~ JSON Serialization and Parsing
- ~ The Internationalization API
- ~ The Console API
- ~ URL APIs
- ~ Timers

=> Iterators and Generators :

- ~ What are Iterators and Generators?
- ~ How Iterators Work?
- ~ Implementing Iterable Objects
- ~ Generators

~ *Advanced Generator Features*

=> Asynchronous JavaScript :

- ~ *What is Asynchronous JavaScript?*
- ~ *Asynchronous Programming with Callbacks*
- ~ *Promises*
- ~ *Async and await*
- ~ *Asynchronous Iteration*

=> Working with Web Browsers :

- ~ *JavaScript in Web Browsers*
- ~ *Web Programming Basics*
- ~ *Events*
- ~ *Scripting Documents*
- ~ *Scripting CSS*
- ~ *Document Geometry and Scrolling*
- ~ *Web Components*
- ~ *SVG: Scalable Vector Graphics*
- ~ *Audio APIs*
- ~ *Location, Navigation, and History*
- ~ *Networking Concepts*
- ~ *Storage*
- ~ *Worker Threads and Messaging*

=> Node js :

- ~ *What is Node.js?*
- ~ *Client-Server Architecture*
- ~ *Single-Threaded Model*
- ~ *Multi-Threaded Model*
- ~ *Multi-Threaded vs Event-Driven*
- ~ *What is Node.js?*
- ~ *Node.js Features*
- ~ *Node.js Installation*
- ~ *Node.js First Example*
- ~ *Blocking vs Non-blocking*
- ~ *Global Objects*
- ~ *File System*
- ~ *Callbacks*
- ~ *Events*
- ~ *Node.js Architecture*
- ~ *NPM(Node Package Manager)*
- ~ *Node.js Modules*
- ~ *Node.js Modules Types*
- ~ *Core Modules*
- ~ *Local Modules*
- ~ *3rd Party Modules*
- ~ *JSON File*
- ~ *Variables*
- ~ *Operators*
- ~ *Functions*
- ~ *Objects*
- ~ *File Systems*
- ~ *Events*
- ~ *HTTP Module*
- ~ *Creating a Web Server using Node.js*
- ~ *Node.js NPM Tutorial*
- ~ *What is NPM?*
- ~ *Main Functions of NPM*
- ~ *Need For NPM*
- ~ *NPM Packages*
- ~ *NPM Installation*
- ~ *JSON File*
- ~ *Node.js Express Tutorial*
- ~ *Introduction to Express.js*
- ~ *Features of Express.js*
- ~ *Getting Started with Express.js*
- ~ *Routing Methods*
- ~ *Building RESTful API with Node.js*
- ~ *What is REST API?*
- ~ *Features of REST API*
- ~ *Principles of REST API*
- ~ *Methods of REST API*
- ~ *Building REST API with Node.js*
- ~ *Contact List MERN App*

=> React JS :

- ~ *Introduction to React*
- ~ *Why should you learn React?*
- ~ *Features of React*
- ~ *React applications*
- ~ *React App & JSX*
- ~ *Functional Components*
- ~ *Applying CSS Styles*
- ~ *Click Events*
- ~ *useState Hook*
- ~ *Lists & Keys*
- ~ *Props & Prop Drilling*
- ~ *Controlled Component Inputs*
- ~ *Project Challenge*
- ~ *useEffect Hook*

- ~ JSON Server
- ~ Fetch API Data
- ~ CRUD Operations
- ~ Fetch Data Challenge
- ~ React Router
- ~ Router Hooks
- ~ Links
- ~ Flexbox Components
- ~ Axios API Requests
- ~ Custom React Hooks
- ~ Context API & useContext Hook
- ~ Build & Deploy Your React Apps

=> Javascript Projects :

- ~ Creating shopping cart app with User Interface

=> Bitcoin Blockchain :

- ~ History of currencies
- ~ Fiat currencies
- ~ Disadvantages of fiat currencies
- ~ Global financial system
- ~ How Central Banks work?
- ~ The 2008 Global Financial Crisis
- ~ Aftermath of 2008 recession
- ~ Creation of Bitcoin- A new decentralised digital currency
- ~ Bitcoin message hash implementation in Javascript
- ~ Immutable ledger practical implementation
- ~ Genesis block
- ~ Timestamp server
- ~ Merkel trees
- ~ Bitcoin as a State Transition System
- ~ Unspent Transaction outputs(UTXOs) Javascript implementation
- ~ Bitcoin whitepaper
- ~ What is a block?
- ~ Components of a Bitcoin block
- ~ Bitcoin Blockchain live implementation
- ~ Distributed Blockchain
- ~ Centralized vs Distributed Blockchain
- ~ consensus mechanism
- ~ Why do we need consensus mechanism in Blockchain networks?
- ~ Byzantine generals problem
- ~ Byzantine fault tolerance- A solution to Byzantine generals problem
- ~ BFT javascript implementation
- ~ Bitcoin nodes
- ~ Bitcoin miners
- ~ Blockchain mining operation
- ~ Mempool
- ~ Bitcoin difficulty adjustment
- ~ Bitcoin halving cycle
- ~ Competing chain problem
- ~ Maintaining immutability - Longest Chain rule
- ~ Block validation
- ~ consensus rules
- ~ Double Spend Validation
- ~ Transaction Input and Output Validation
- ~ Coinbase Transaction Reward Validation
- ~ Coinbase Maturity
- ~ Coinbase Transaction Block Height
- ~ Signature Check Counting
- ~ SigChecks
- ~ Mining incentive
- ~ Mining optimized hardware
- ~ CPU processing power
- ~ GPUs for mining
- ~ Application Specific Integrated Circuits(ASIC) miners
- ~ CPU vs GPU vs ASIC miners
- ~ Distributed peer to peer Blockchain live implementation
- ~ Distributed peer to peer Blockchain Javascript implementation
- ~ Token transaction live implementation using distributed peer to peer blockchain
- ~ Coinbase transaction live implementation using distributed peer to peer blockchain
- ~ Token and Coinbase transaction Javascript implementation
- ~ Bitcoin public key and private key
- ~ Public key and private key generation
- ~ Bitcoin addresses
- ~ Bitcoin digital signatures
- ~ Signing a peer to peer message with private key- Javascript implementation
- ~ Verifying peer to peer message using public key and digital signature-implementation
- ~ Signing and verifying currency transaction- implementation
- ~ Complete Bitcoin Blockchain implementation with transaction signatures

=> Probable attacks in Bitcoin blockchain :

- ~ Sybil Attack
- ~ Race Attack
- ~ Finney Attack
- ~ Vector76 Attack
- ~ 51% Attack

=> Bitcoin Project :

- ~ Building a Blockchain using Javascript

=> Ethereum Blockchain :

- ~ Module overview
- ~ Understanding the drawbacks of Bitcoin blockchain
- ~ Lack of Turing-completeness
- ~ Value-blindness
- ~ Lack of state
- ~ Blockchain-blindness
- ~ Origin of Ethereum- The programmable currency
- ~ The Decentralized Applications revolution and modern state of blockchain systems
- ~ Decentralized Applications vs Centralized Applications
- ~ Ethereum Accounts overview
- ~ Contract Accounts(CA)
- ~ Externally Owned Accounts(EOA)
- ~ Fields in Ethereum accounts
- ~ Ethereum Account messages
- ~ Ethereum Account transactions
- ~ Ethereum Addresses
- ~ Units of Ether
- ~ Ether Gas
- ~ Computing total gas cost for Ethereum transactions
- ~ Ethereum gas price Javascript implementation
- ~ Ethereum as a State Transition Function
- ~ Ethereum Architecture
- ~ Ethereum Virtual Machine(EVM)
- ~ EVM nodes vs mining nodes
- ~ EVM Bytecode
- ~ EVM Instruction Set
- ~ EVM Opcode
- ~ EVM Storage
- ~ EVM Memory
- ~ EVM Stack
- ~ Geth setup and EVM practical
- ~ Converting bytecode to opcode
- ~ Application Binary Interface(ABI)
- ~ Understanding end-to-end Ethereum Blockchain transaction in Javascript
- ~ Ethereum Smart Contracts architecture

=> Ethereum 2.0 :

- ~ Why was Ethereum 2.0 proposed?
- ~ Energy usage in Proof of Work
- ~ Gas costs in Ethereum 1.0
- ~ Potential scalability issues
- ~ Moving from Proof of Work to Proof of Stake
- ~ Proof of Stake in Ethereum 2.0
- ~ Validators
- ~ Staking
- ~ Attestation
- ~ Crosslinks
- ~ Finality
- ~ consensus clients
- ~ Execution clients
- ~ Sharding
- ~ Shard chains
- ~ Beacon chain
- ~ Data rollup in Ethereum 2.0
- ~ Forking in Blockchain
- ~ Hard Fork
- ~ Soft Fork
- ~ The DAO attack and Ethereum Hard Fork

=> Solidity :

- ~ What is Solidity?
- ~ Why should you learn Solidity programming?
- ~ Introduction to Smart Contracts
- ~ Solidity Installation
- ~ Remix IDE
- ~ Installing Solidity in npm / Node.js
- ~ Layout of a Solidity Source File
- ~ SPDX License Identifier
- ~ Pragmas
- ~ Comments in Solidity
- ~ Structure of a Smart Contract

=> Solidity Value Types :

- ~ Solidity datatypes
- ~ Booleans
- ~ Integers
- ~ Address Type
- ~ Address Literals
- ~ Contract Types
- ~ Byte Type
- ~ String Types
- ~ Enums in Solidity

=> Solidity Reference Types :

- ~ Data locations- storage, memory and callback
- ~ Solidity Arrays
- ~ Fixed Arrays
- ~ Dynamic Arrays

- ~ *Bytes and Strings as Arrays*
- ~ *Array Slicing*
- ~ *Structs*
- ~ *Mapping Types*

=> Solidity Units and Global Variables :

- ~ *Ether Units*
- ~ *Time Units*

=> Solidity Control Structures :

- ~ *If statement*
- ~ *If/else statement*
- ~ *Nested if/else statements*
- ~ *Solidity Loops*
- ~ *For loop*
- ~ *While loop*
- ~ *Do-while loop*
- ~ *Break statement*
- ~ *Continue statement*

=> ABI Encoding and Decoding Functions :

- ~ *ABI encoder*
- ~ *ABI decoder*

=> Cryptographic Functions :

- ~ *Keccak256*
- ~ *SHA256*
- ~ *Ripemd160*
- ~ *Ecrecover*

=> Smart Contracts :

- ~ *Creating Smart Contracts*
- ~ *Constructor*
- ~ *Scope visibility*
- ~ *State variable visibility*
- ~ *Functions*
- ~ *Function visibility*
- ~ *Getter functions*
- ~ *Setter functions*
- ~ *Function modifiers*
- ~ *Return variables and returning multiple values*
- ~ *Immutable state variables*
- ~ *Payable functions*
- ~ *Fallback functions*
- ~ *View functions*
- ~ *Pure functions*
- ~ *Function overloading*
- ~ *Function overriding*
- ~ *Solidity Events*
- ~ *Block and Transaction details*
- ~ *Solidity Inheritance*
- ~ *Single Inheritance*
- ~ *Multiple Inheritance*
- ~ *Heirarchical Inheritance*
- ~ *Multilevel Inheritance*
- ~ *Abstract Contracts*
- ~ *Solidity Interfaces*
- ~ *Solidity Libraries*

=> Solidity Programming Applications :

- ~ *Ether Wallet*
- ~ *Multi Sig Wallet*
- ~ *Iterable Mapping*
- ~ *ERC20*
- ~ *ERC721*
- ~ *Uni-directional Payment Channel*
- ~ *Bi-directional Payment Channel*
- ~ *NFT Auction*
- ~ *Crowd Fund*
- ~ *Time Lock*

=> Common Ethereum Blockchain Hacks and Loopholes :

- ~ *Re-Entrancy Attack*
- ~ *Self Destruct*
- ~ *Accessing Private Data*
- ~ *Denial of Service*
- ~ *Phishing with tx.origin*
- ~ *Hiding Malicious Code with External Contract*
- ~ *Honeypot*
- ~ *Front Running*
- ~ *Block Timestamp Manipulation*
- ~ *Signature Replay*
- ~ *Bypass Contract Size Check*

=> Introduction to Blockchain Development Frameworks :

- ~ *Introduction to Smart Contract Development in Production*
- ~ *Web3 libraries for Javascript*
- ~ *Smart Contract development tools*
- ~ *Web3 Providers*
- ~ *Wallets*

=> Truffle Suite :

- ~ Truffle overview
- ~ Truffle Installation
- ~ Creatin a new project in Truffle
- ~ Exploring project directories in Truffle
- ~ Compiling Smart Contracts
- ~ Building Artifacts
- ~ Handling Dependencies
- ~ Reading and writing Smart Contract data
- ~ Smart Contract Transactions in Truffle
- ~ Function calls in Truffle
- ~ Abstractions
- ~ Executing Contract functions
- ~ Making Transactions
- ~ Processing Transaction results
- ~ Catching events
- ~ Add a new contract to the network
- ~ Sending ether to a contract
- ~ Invoking overloaded methods
- ~ Using enumerations
- ~ Preserving Files and Content to Storage Platforms
- ~ Inter Planetary File System(IPFS)
- ~ Filecoin
- ~ Textile Buckets
- ~ Running Migrations
- ~ Initial Migration
- ~ Truffle Deployer
- ~ Network considerations
- ~ Truffle Deployer API
- ~ Integrating Truffle with Metamask
- ~ Using Truffle Dashboard
- ~ Using truffle Debugger
- ~ Truffle Develop and Truffle Console
- ~ Writing and executing external scripts
- ~ Testing Smart Contracts
- ~ Writing Automated Tests in Javascript
- ~ Writing Automated Tests in Solidity
- ~ Truffle Build Process
- ~ Truffle Boxes
- ~ Ethereum Name Service
- ~ Truffle Event System
- ~ Network Configuration and Dapp Deployment
- ~ Ganache- Ethereum Client for Truffle Suite
- ~ Installing Ganache
- ~ Ganache Workspaces
- ~ Ganache Ethereum Workspace
- ~ Understanding Workspace Default Configuration in Ganache
- ~ Managing Ganache configurations and settings
- ~ Configuring Truffle to connect to Ganache
- ~ Managing Truffle projects in Ganache
- ~ Exploring the Contracts page
- ~ Exploring the Transactions page
- ~ Linking and unlinking a Truffle project
- ~ Ganache Workspaces
- ~ Creating Workspaces
- ~ Deleting Workspaces
- ~ Editing Workspaces
- ~ Ethereum Workspace
- ~ Loading Existing Workspaces
- ~ Switching Workspaces

=> Hardhat :

- ~ Introduction To Hardhat - Ethereum development environment for professionals
- ~ Hardhat Installation
- ~ Creating a Hardhat project
- ~ Configuring Ethereum Networks
- ~ Configuring the compiler
- ~ Compiling your contracts
- ~ Artifacts
- ~ Writing deployment scripts
- ~ Deploying the Contracts
- ~ Testing Smart Contracts
- ~ Running tests with Ganache
- ~ Running tests on Visual Studio Code
- ~ Running multiple tests in parallel
- ~ Running tasks
- ~ Hardhat Console
- ~ Creating custom tasks
- ~ Hardhat Runtime Environment(HRE)
- ~ Hardhat Plugins
- ~ Optimizing Plugins
- ~ Verbose Logging for debugging
- ~ Solutions to common runtime problems

=> Web3.js :

- ~ Introduction to Web3.js
- ~ Why should you learn Web3.js?
- ~ Applications of Web3.js

- ~ Installing Web3.js using NPM
- ~ Web3 modules
- ~ Creating a new Web3 instance
- ~ Introduction to Web3 Providers
- ~ Setting up a Web3 Provider
- ~ Batch request
- ~ Extending Web3 modules
- ~ Introduction to Web3.eth
- ~ Checksum addresses overview
- ~ Fetching default blockchain details
- ~ Transaction methods
- ~ Block Node methods
- ~ Subscriber Methods
- ~ Web3.js Smart Contract objects and methods
- ~ User wallet and account methods
- ~ Interacting with Ethereum node accounts using web3.eth.personal
- ~ Working with ABI in web3.js
- ~ Commonly used utilities in web3.js
- ~ Hardhat automated testing with Web3.js and Truffle

=> Ethers.js :

- ~ What is Ethers?
- ~ Ethers.js Features
- ~ Installing Ethers.js using NPM
- ~ Connecting to Ethereum: MetaMask
- ~ Connecting to Ethereum: RPC
- ~ Building blocks of Ethers.js- Signers, Providers and Contracts

=> Ethers.js Providers :

- ~ What are Providers?
- ~ Ethers.js provider API overview
- ~ Provider Account methods
- ~ Blocks Methods
- ~ Ethereum Naming Service (ENS) Methods
- ~ EnsResolver
- ~ Logs Methods
- ~ Network Status Methods
- ~ Transactions Methods
- ~ Event Emitter Methods
- ~ Inspection Methods
- ~ BaseProvider
- ~ JsonRpcProvider
- ~ JsonRpcSigner
- ~ JsonRpcUncheckedSigner
- ~ StaticJsonRpcProvider
- ~ Node-Specific Methods
- ~ API Providers
- ~ EtherscanProvider
- ~ InfuraProvider
- ~ AlchemyProvider
- ~ CloudflareProvider
- ~ PocketProvider
- ~ AnkrProvider
- ~ Other Providers
- ~ FallbackProvider
- ~ IpcProvider
- ~ JsonRpcBatchProvider
- ~ UrlJsonRpcProvider
- ~ Web3Provider
- ~ WebSocketProvider

=> Ethers.js Signers :

- ~ What are Signers?
- ~ Wallet Signer
- ~ JsonRPC Signer
- ~ Signer class and member functions
- ~ Ethers.js Wallet class and member functions
- ~ VoidSigner
- ~ Interacting with Externally Owned Accounts (EOA)

=> Smart Contract Interaction :

- ~ Creating new Smart Contract instance
- ~ Contract Properties
- ~ Contract Methods
- ~ Events
- ~ ContractFactory
- ~ Creating ContractFactory Instances
- ~ ContractFactory Interface Properties
- ~ ContractFactory Methods
- ~ Meta-Class
- ~ Deploying a Contract
- ~ Connecting to a Contract
- ~ Properties
- ~ Methods
- ~ Events
- ~ Meta-Class Methods
- ~ Meta-Class Filters
- ~ Hardhat automated testing with Ether.js and Waffle

=> Ethereum Blockchain Projects :

- ~ Building cryptocurrency with ICO
- ~ Building decentralized ecommerce website
- ~ Building decentralized voting application
- ~ Decentralized music sharing app
- ~ Token contract swap application
- ~ Full stack email dapp

=> Oracles :

- ~ What is a Blockchain Oracle?
- ~ Solving the Oracle problem
- ~ Decentralized Oracles
- ~ Types of Blockchain Oracles
- ~ Applications of Blockchain oracles

=> Chainlink overview :

- ~ Introduction to Chainlink
- ~ Understanding the Chainlink Ecosystem
- ~ Chainlink Features
- ~ Chainlink Applications as Decentralized Oracles
- ~ Chainlink Architecture
- ~ ERC677 Standard
- ~ The LINK token
- ~ Decentralized Data Model
- ~ Chainlink Off-chain Reporting
- ~ Chainlink Whitepaper

=> Data Feeds :

- ~ Introduction to Data Feeds
- ~ Using Data Feeds
- ~ Fetchin Historical Cryptocurrency Price Data
- ~ Chainlink Feed Registry
- ~ Using ENS with Data Feeds
- ~ Contract Addresses
- ~ Ethereum Data Feeds
- ~ Binance Smart Chain Data Feeds
- ~ Polygon (Matic) Data Feeds
- ~ Gnosis Chain (xDai) Data Feeds
- ~ HECO Chain Data Feeds
- ~ Avalanche Data Feeds
- ~ Fantom Data Feeds
- ~ Arbitrum Data Feeds

=> Chainlink VRF :

- ~ Introduction to Chainlink VRF(Verifiable Random Function)
- ~ Applications of randomness in Blockchain
- ~ Generating randomness
- ~ Some security considerations in Chainlink VRF
- ~ Smart Contract Integration

=> Custom Data Feeds :

- ~ Using any API
- ~ Make a GET Request
- ~ Multi-Variable Responses
- ~ Large Responses
- ~ Make an Existing Job Request
- ~ Find Existing Jobs
- ~ Contract Addresses

=> Chainlink Keepers :

- ~ Automating Smart Contracts
- ~ Introduction to Chainlink Keepers
- ~ Keepers Architecture
- ~ Keepers-compatible Contracts
- ~ Register an Upkeep
- ~ Manage your Upkeeps
- ~ Utility Contracts
- ~ EthBalanceMonitor
- ~ Supported Networks
- ~ Chainlink Keepers Economics

=> Oracle Projects :

- ~ Live cryptocurrency trading using chainlink
- ~ Insurance Dapp using chainlink

=> The Graph :

- ~ The Graph Protocol
- ~ The Graph architecture
- ~ Edge and Node
- ~ Everest Registry
- ~ Graph Protocol
- ~ The Graph vs Etherscan
- ~ Graph-cli Installation
- ~ Creating new subgraphs
- ~ Writing subgraphs
- ~ Publishing a Subgraph to the Decentralized Network

=> GraphQL API :

- ~ Queries
- ~ Sorting
- ~ Pagination
- ~ Filtering

- ~ Time-travel queries
- ~ Fulltext Search Queries
- ~ Validation
- ~ Schema
- ~ Entities
- ~ Signalling
- ~ Curation
- ~ Delegators
- ~ Consumers
- ~ Deploying subgraphs
- ~ Subgraph logging
- ~ Graph protocol testnet using docker compose
- ~ Ethereum node monitoring using The Graph, Prometheus and Grafana

=> The Graph Networking :

- ~ Introduction to indexers
- ~ Revenue streams
- ~ Distribution
- ~ Allocation life cycles
- ~ Querying and indexing subgraphs
- ~ IPFS Hash convertor

=> AssemblyScript API for The Graph :

- ~ Installing AssemblyScript API
- ~ API Reference
- ~ Versions
- ~ Built-in Types
- ~ Store API
- ~ Ethereum API
- ~ Logging API
- ~ IPFS API
- ~ Crypto API
- ~ JSON API
- ~ Type Conversions Reference
- ~ Data Source Metadata
- ~ Entity and DataSourceContext

=> The Graph Unit Testing :

- ~ Installing dependencies
- ~ WSL (Windows Subsystem for Linux)
- ~ Usage
- ~ CLI options
- ~ Docker
- ~ System Configuration
- ~ Demo subgraph
- ~ Asserts
- ~ Writing a Unit Test
- ~ Common test scenarios
- ~ Hydrating the store with a certain state
- ~ Calling a mapping function with an event
- ~ Calling all of the mappings with event fixtures
- ~ Mocking contract calls
- ~ Asserting the state of the store
- ~ Interacting with Event metadata
- ~ Asserting variable equality
- ~ Asserting that an Entity is not in the store
- ~ Printing the whole store (for debug purposes)
- ~ Expected failure
- ~ Logging
- ~ Testing derived fields
- ~ Testing dynamic data sources
- ~ Test Coverage

=> Project :

- ~ Building a Full-stack Blockchain Application using Ethereum, Polygon, Next.js and GraphQL

=> Decentralized Autonomous Organisations(DAO) :

- ~ What are DAOs?
- ~ Why do we need DAOs?
- ~ DAO membership
- ~ Token-based membership
- ~ Share-based membership
- ~ How do DAOs work?
- ~ Properties of DAOs
- ~ Ethereum and DAOs
- ~ Understanding Governance Mechanisms
- ~ DAOs and the principal-agent problem
- ~ Building Decentralized Autonomous Organisations
- ~ Defining the DAO purpose
- ~ Building the DAO voting mechanism
- ~ Creating the governance token
- ~ DAO fund management
- ~ Initial Coin Offering (ICO)
- ~ Creating a DAO on Aragon
- ~ Creating a DAO using Snapshot
- ~ Building a DAO using DAOstack Alchemy

=> Creating a Custom DAO Project :

- ~ Understanding custom DAOs
- ~ Finding the purpose for our Custom DAO

- ~ Designing the voting architecture
- ~ Implementing the voting architecture in Solidity
- ~ Designing the components of the governance token(DAO cryptocurrency)
- ~ Creating the governance token in Solidity
- ~ Fund Management for our custom DAO
- ~ Designing the Multi-signature wallet for Fund Management
- ~ Creating the Multi-signature wallet in Solidity
- ~ Testing DAO Smart Contracts
- ~ Deploying the DAO to testnet

=> Decentralized Finance(DeFi) :

- ~ The Traditional Financial Institutions
- ~ Centralization & Transparency
- ~ The Banks
- ~ General Public Accessibility
- ~ Decentralized Finance
- ~ The DeFi Ecosystem
- ~ How does DeFi work?
- ~ DeFi Categories
- ~ Decentralized Stablecoins
- ~ Lending and Borrowing
- ~ Decentralized Exchanges
- ~ Derivatives
- ~ Fund Management
- ~ Lottery
- ~ Decentralized payments systems
- ~ Insurance
- ~ Yield Farming
- ~ Liquidity Mining
- ~ Airdrops
- ~ Decentralized Prediction Markets

=> Famous DeFi Protocols :

- ~ Aave
- ~ yEarn
- ~ Compound
- ~ Uniswap
- ~ Sushiswap
- ~ Maker
- ~ Numerai
- ~ Curve Finance
- ~ Alpha Finance

=> DeFi projects :

- ~ Understanding DeFi Project Architecture
- ~ Components of Full-Stack DeFi applications
- ~ Designing DeFi project workflows
- ~ Building a Decentralized lottery system
- ~ Building a Decentralized borrowing and lending platform
- ~ Building a Decentralized stablecoin

=> Non Fungible Tokens (NFT) :

- ~ What is a Non Fungible Token(NFT)?
- ~ How does a NFT work?
- ~ Fungible Tokens vs Non Fungible Tokens
- ~ Exploring uses of NFTs
- ~ NFT as an internet of assets
- ~ NFT as a store of value
- ~ The Metaverse and NFT's role in it

=> NFT Platforms :

- ~ What are NFT Platforms/Marketplaces?
- ~ CryptoKitties
- ~ Opensea
- ~ Rarible
- ~ Decentraland
- ~ Binance NFT
- ~ Enjin Marketplace
- ~ Axie Marketplace
- ~ Foundation
- ~ Nifty Gateway
- ~ Mintable
- ~ Theta Drop

=> NFT Transaction Fees :

- ~ Gas Fees in NFT
- ~ What are one-time Gas Fees NFT?
- ~ Recurring Gas Fees
- ~ Actions in Gas Fees
- ~ Check Ethereum Gas Fee
- ~ Create and Sell NFTs without Gas Fees
- ~ NFT Marketplace Fees

=> NFT Programming :

- ~ Getting data for generating NFTs
- ~ Assigning trait rarity for digital assets
- ~ Classifying traits
- ~ Defining image traits
- ~ Validating uniqueness
- ~ Trait Counting
- ~ Generate the Images

- ~ Understanding NFT metadata
- ~ Uploading NFT images to IPFS
- ~ Generate NFT metadata
- ~ Upload the metadata to IPFS
- ~ Environment Setup for Smart Contract deployment
- ~ Creating Alchemy account
- ~ Writing NFT smart contract
- ~ Integrating Metamask, Alchemy and your Project

=> NFT project :

- ~ Building a complete NFT Marketplace with User Interface

=> Polygon Blockchain(MATIC) :

- ~ Introduction to Polygon Blockchain
- ~ Why should you use Polygon network?
- ~ Layer 1 vs Layer 2 Blockchains
- ~ Features of Polygon Blockchain
- ~ Polygon Architecture
- ~ Zero-Knowledge cryptography
- ~ Zero-Knowledge rollups

=> Polygon Network :

- ~ Introduction to Polygon Mainnet and Testnet
- ~ Mapped Tokens
- ~ Matic Gas Token
- ~ Genesis Contracts
- ~ Minimum Technical Requirements
- ~ Snapshot Instructions for Heimdall and Bor
- ~ Full Node Binaries
- ~ Full Node Deployment
- ~ Polygon Wallets
- ~ Arkane
- ~ Formatic
- ~ Metamask

=> Polygon-Ethereum Bridge :

- ~ Introduction to Polygon POS bridge
- ~ Matic.js
- ~ Installing matic.js using NPM
- ~ Polygon Web3.js Setup
- ~ Polygon Ethers.js Setup
- ~ Supported libraries
- ~ Web3js setup
- ~ Ethers setup
- ~ Matic.js POS
- ~ Matic.js POSClient
- ~ Matic.js ERC20
- ~ Matic.js ERC721
- ~ Matic.js ERC1155
- ~ isCheckPointed
- ~ isDeposited
- ~ deposit ether
- ~ FxPortal
- ~ Set ProofApi

=> Advanced Concepts :

- ~ ABIManager
- ~ Plugins
- ~ ExitUtil
- ~ PoS Bridge
- ~ Using Polygon Edge
- ~ Instantiating Polygon Edge
- ~ Deposit and Checkpoint Event Tracking
- ~ Deployment Details
- ~ Mapping Assets using POS
- ~ Tools
- ~ Wallet Widget
- ~ Submit Mapping Request
- ~ Polygon Mintable Assets
- ~ IPFS - Filecoin
- ~ Using IPFS
- ~ Using Filecoin
- ~ Mint with NFT.storage on Polygon
- ~ Polygon with Oracles
- ~ Chainlink integration

=> Polygon Projects :

- ~ Retail supply chain Application using Polygon Network
- ~ Building a Social media Dapp on Polygon

=> Polkadot :

- ~ Polkadot Overview
- ~ Polkadot Whitepaper
- ~ Polkadot Architecture
- ~ Parachains
- ~ Parathreads
- ~ Substrate Installation

=> Substrate Fundamentals :

- ~ Runtime environment and setup
- ~ Extrinsic

- ~ *Account Abstractions*
- ~ *Transaction Pool*
- ~ *Session Keys*
- ~ *Transaction Weight*
- ~ *Execution*
- ~ *Off-Chain Features*

=> Runtime Development :

- ~ *Frames*
- ~ *Macros*
- ~ *Metadata*
- ~ *Storage*
- ~ *Origins*
- ~ *Events and Errors*
- ~ *Weights and Fees*
- ~ *Benchmarking*
- ~ *Debugging*
- ~ *Testing*
- ~ *Randomness*
- ~ *Chain Specification*
- ~ *Upgrades*
- ~ *Pallet Coupling*
- ~ *Custom RPCs*
- ~ *Smart Contract Toolkits*

=> Development Integration :

- ~ *Polkadot-JS*
- ~ *Client Libraries*
- ~ *Substrate Connect*

=> Development Tools :

- ~ *SR tool*
- ~ *Subxt*
- ~ *Tx Wrapper*
- ~ *Sub Flood*
- ~ *Substrate Archive*
- ~ *Sidecar*
- ~ *Polkadot Launch*

=> Advanced topics in Polkadot :

- ~ *Account Info*
- ~ *SCALE Codec for Substrate*
- ~ *Consensus*
- ~ *Block Import*
- ~ *Executor*
- ~ *Cryptography*
- ~ *Storage*
- ~ *SS58 Address Format*
- ~ *Hash Collections*

Manual Testing Kickstart

Topic Name : TESTING

Sub-topic Name : MANUAL TESTING

Course link : <https://ineuron.ai/course/Manual-Testing-Kickstart>

Course Description :-

The goal of this course is to learn the fundamental ideas and methods of Software Testing. Software testing Basics, SDLC Models, Waterfall, V, Spiral, and Agile model, STLC, Phases & Types of Testing, Black Box & White Box Testing, Smoke & Sanity Testing, Regression & Retesting will all be covered in the training. You will learn about, Test Planning, Test case identification & Creation, Test case execution, Bug Management, and Bug Tracking tools like Jira, and Test Reporting. MySQL for backend data validation will also be covered.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Assignments
- => Course completion certificate

What you will learn :-

- => Basics of Testing
- => Principles Of Testing
- => SDLC
- => Waterfall
- => Spiral
- => Test Estimation & Test Management
- => TestCase Development
- => Testing Types
- => Testing Techniques

Requirements :-

- => Knowledge of C++
- => System with Internet Connection
- => Interest to learn
- => Dedication

Instructors :-

=> Kiran Sahu :

~ QA Manager with 12+ years of professional experience, worked in Brands like Infosys, Delhivery, Mydala, Aurea, Jive, Crossover, Agama Solutions & OSTC, have experience of working in global platforms and with multinational professionals. Strong domain knowledge on Retail, Logistics, Banking, Trading, Ecommerce Applications. Experience in Training and Mentoring Candidates all across the globe on Software Testing, MySQL and Agile.

Curriculum details :-

=> Introduction to Manual Testing :

- ~ Basics of Testing
- ~ Principles Of Testing

=> Manual Testing :

- ~ SDLC
- ~ STLC SDLCvsSTLC
- ~ Waterfall
- ~ SoftwareTesting V
- ~ Spiral
- ~ Agile
- ~ Test Estimation & Test Management
- ~ TestPlan
- ~ TestCase Development
- ~ RTM

=> Manual Testing Phases :

- ~ Phases of testingUnitIntegration
- ~ Phases of testingSystem Testing
- ~ Phases of TestingUAT

=> Testing Types :

- ~ Types Of Testing SmokeandSanity
- ~ Types Of Testing Regression vs Retesting
- ~ Types Of Testing Func NonFunc Testing
- ~ Types Of Testing WhiteBox BlackBox Testing

=> Testing Techniques :

- ~ Testing Techniques Equivalence Partitioning And BVA
- ~ Testing TechniquesDecisionTable

=> Manual Testing MySQL :

- ~ Mysql create insertselect
- ~ Mysql orderby
- ~ Mysql groupby
- ~ Mysql wildcards
- ~ Mysql nullalter drop rename
- ~ Mysql limitddldmldcl
- ~ Mysql keys
- ~ Mysql definingKeys
- ~ Mysql join
- ~ Mysql subquery

Reinforcement Learning

Topic Name : DATA SCIENCE

Sub-topic Name : REINFORCEMENT LEARNING

Course link : <https://ineuron.ai/course/Reinforcement-Learning>

Course Description :-

You will learn the basics of reinforcement learning through this course. Adaptive learning systems are needed for artificial intelligence to reach its full potential. Implementing a complete RL solution will teach you how Reinforcement Learning (RL) solutions assist in solving real-world problems through trial-and-error interaction.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => Basics of Reinforcement Learning
- => Foundations of Reinforcement Learning
- => OpenAI GYM Cartpole Experiment

Requirements :-

- => System with minimum i3 processor or better
- => At least 4 GB of RAM
- => Working internet connection
- => Dedication to learn

Instructors :-

=> Sunny Bhaveen Chandra :

~ Sr. Data Scientist and lecturer at iNeuron.ai with working experience in computer vision, natural language processing and embedded systems. Hands-on experience leveraging machine learning, deep learning, transfer learning models to solve challenging business problems. Also, he has a vast interest in Robotics.

Curriculum details :-

- => Introduction to Reinforcement Learning :
 - ~ Introduction to the course
 - ~ RL Introduction Part 1
 - ~ RL Introduction Part 2
 - ~ OpenAI GYM Cartpole Experiment

Industry Ready Data Science Projects Tech Neuron

Topic Name : DATA SCIENCE

Sub-topic Name : MACHINE LEARNING PROJECT

Course link : <https://ineuron.ai/course/Industry-Ready-Data-Science-Projects-Tech-Neuron>

Course Description :-

Ready to use end-to-end data science projects for real-world business use cases. We will be discussing projects from very scratch such as understanding problem statements, capturing requirements, and various aspects of project design using different documentation such as High-Level Design, Low-Level Design, and Architecture Design. Practical use of MLOPS practices using tools such as MLFLOW, Wandb. Pipeline implementation for training, retraining, and inferencing. Designing dashboard to present important KPIs to monitor system and model performance and generate alert to notify the appropriate parties to address serious problems if it is about to occur.

Course Features :-

- => Online Instructor-led learning
- => Doubt Clearing
- => Proper Roadmap for building AI projects
- => Lifetime Dashboard access
- => Recording of Live Class
- => Material
- => Interview Questions
- => Resume Building
- => Career Guidance
- => Quiz in every module - Based on Real Time Questions
- => Certificate
- => Industry Level Projects and Case studies
- => Capstone Projects

What you will learn :-

- => System Architecture
- => High Level Design
- => Component Selection
- => Low Level Design
- => Core utility design
- => Deployment Architecture
- => Multistage pipeline for CI/CD
- => ML Pipeline Understanding
- => Training Pipeline Implementation
- => Inference Pipeline Implementation
- => Retraining Pipeline Implementation
- => Deployment of ML Pipeline on Cloud
- => Monitoring of System and Model Performance

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication
- => Knowledge of Python
- => Knowledge of Machine Learning

Instructors :-

=> Avnish Yadav :

~ 3+ years of experience in various domains such as data scientist, data analyst, database developer, and .net developer. Implemented various sophisticated business requirements, performed an analysis of various data to capture insights and hidden patterns. Fine and tuned various regression and classification-based algorithms for prediction. Implemented various ETL pipelines to fulfil the business requirement. Automated various machine learning pipelines such as data loading, data cleaning, data validation, model selection, model tuning, and model monitoring pipeline. Implemented

machine learning pipeline in azure machine learning studio. I have a keen interest to solve complicated machine learning problems to fulfil business requirements.

=> Ketan Gangal :

~ I have worked in data science for more than two years, and I have a track record of successfully implementing data science pipelines in production with practical expertise using ML-Ops, deep learning & machine learning. I also love sequence processing because it is deeply inspired by humans as our feeling, thoughts, emotions, sensations, language are sequential in nature if we can enable machine to understand sequence of information and act accordingly we can make significant progress towards true artificial intelligence.

=> krish naik :

~ Having 10+ years of experience in Data Science and Analytics with product architecture design and delivery. Worked in various product and service based Company. Having an experience of 5+ years in educating people and helping them to make a career transition.

=> Sudhanshu Kumar :

~ Having 8+ years of experience in Big data, Data Science and Analytics with product architecture design and delivery. Worked in various product and service based Company. Having an experience of 5+ years in educating people and helping them to make a career transition.

Curriculum details :-

=> Project - Sensor Fault Detection :

- ~ Project Introduction
- ~ Project Business Use case
- ~ System Architecture
- ~ High Level Design
- ~ Component Selection
- ~ Low Level Design
- ~ Core utility design
- ~ Deployment Architecture
- ~ Multistage pipeline for CI/CD
- ~ Technology Stack
- ~ Python, Pandas, Sklearn, Mlflow, Cloud, Prometheus and Grafana, Docker, RDBMS, Cloud Storage, Flask, Git, GitHub
- ~ ML Pipeline Understanding
- ~ Type of ML Pipeline
- ~ Training Pipeline
- ~ Inferencing Pipeline
- ~ Retraining Pipeline
- ~ Training Pipeline Implementation
- ~ Introduction to Training Pipeline
- ~ Data Ingestion From Data Source
- ~ Data Validation
- ~ EDA, Data Preprocessing, Feature Engineering Model Selection
- ~ Customize Model Training
- ~ Model Training, Selection and Hyperparameter Tuning
- ~ Model Analysis and Evaluation
- ~ Model Push/ Export
- ~ Inference Pipeline Implementation
- ~ Introduction to Inference Pipeline
- ~ Understanding of the use of Artifact Generated by Training Pipeline
- ~ Data Validation
- ~ Data Preprocessing and Feature Engineering
- ~ Prediction using preprocessed data
- ~ Retraining Pipeline Implementation
- ~ Introduction to Retraining Pipeline
- ~ Model Analysis and Performance Monitoring of Prediction Pipeline
- ~ Creating Trigger to Initiate Model Retraining
- ~ Deployment of ML Pipeline on Cloud
- ~ Schedule and Orchestrate Training Pipeline
- ~ Deployment of Inference Pipeline as an API
- ~ Deployment of Retraining Pipeline
- ~ Monitoring of System and Model Performance
- ~ Importance of Monitoring
- ~ Visualization of KPI and Other Indicator
- ~ System and Model Performance Visualization
- ~ Implementation of Alert and Notification to prevent Failure
- ~ Project Conclusion

=> Project - Financial Product Complaint :

- ~ Project Introduction
- ~ Project Business Use case
- ~ System Architecture
- ~ High Level Design
- ~ Component Selection
- ~ Low Level Design
- ~ Core utility design
- ~ Deployment Architecture
- ~ Multistage pipeline for CI/CD
- ~ Technology Stack
- ~ Python, Pytorch, Cloud, Prometheus and Grafana, Docker, RDBMS, Cloud Storage, Flask, Git, GitHub
- ~ ML Pipeline Understanding
- ~ Type of ML Pipeline
- ~ Training Pipeline
- ~ Inferencing Pipeline
- ~ Retraining Pipeline
- ~ Training Pipeline Implementation
- ~ Introduction to Training Pipeline
- ~ Data Ingestion From Data Source
- ~ Data Validation
- ~ EDA, Data Preprocessing, Feature Engineering Model Selection
- ~ Model Training, Selection and Hyperparameter Tuning
- ~ Model Analysis and Evaluation

- ~ Model Push/ Export
- ~ Inference Pipeline Implementation
- ~ Introduction to Inference Pipeline
- ~ Understanding of the use of Artifact Generated by Training Pipeline
- ~ Data Validation
- ~ Data Preprocessing and Feature Engineering
- ~ Prediction using preprocessed data
- ~ Retraining Pipeline Implementation
- ~ Introduction to Retraining Pipeline
- ~ Model Analysis and Performance Monitoring of Prediction Pipeline
- ~ Creating Trigger to Initiate Model Retraining
- ~ Deployment of ML Pipeline on Cloud
- ~ Schedule and Orchestrate Training Pipeline
- ~ Deployment of Inference Pipeline as an API on Elastic Container Serving
- ~ Deployment of Retraining Pipeline
- ~ Monitoring of System and Model Performance
- ~ Importance of Monitoring
- ~ Visualization of KPI and Other Indicator
- ~ System and Model Performance Visualization
- ~ Implementation of Alert and Notification to prevent Failure
- ~ Project Conclusion

=> Project - Face Authenticator :

- ~ Project Introduction
- ~ Project Business Use case
- ~ System Architecture
- ~ High Level Design
- ~ Component Selection
- ~ Low Level Design
- ~ Core utility design
- ~ Deployment Architecture
- ~ Multistage pipeline for CI/CD
- ~ Technology Stack
- ~ Python, MongoDB, Deepface, Flask, Docker, EC2 Instance, Git, Github, SQL
- ~ Face Authenticator Pipeline
- ~ Understanding Face Authenticator mechanism
- ~ Face Registration Pipeline
- ~ Face Identification Pipeline
- ~ Face Registration Pipeline
- ~ Capturing Images of a Person
- ~ Generating Embedding of Facial Image
- ~ Save Embedding in Database
- ~ Face Identification Pipeline
- ~ Detecting face of a Person at login portal
- ~ Generate embedding of captured face
- ~ Search Generated Embedding in DB using similarity metrics Triplet Loss
- ~ Monitoring of System and Model Performance
- ~ Importance of Monitoring
- ~ Visualization of KPI and Other Indicator
- ~ System and Model Performance Visualization
- ~ Implementation of Alert and Notification to prevent Failure
- ~ Project Conclusion

=> Project - Embedding based search engine :

- ~ Project Introduction
- ~ Project Business Use case
- ~ System Architecture
- ~ High Level Design
- ~ Component Selection
- ~ Low Level Design
- ~ Core utility design
- ~ Deployment Architecture
- ~ Multistage pipeline for CI/CD
- ~ Technology Stack
- ~ Python, Pytorch, Hugging Face, Transformer, Prometheus and Grafana, Docker, RDBMS, Cloud Storage, Flask, Git, GitHub
- ~ ML Pipeline Understanding
- ~ Type of ML Pipeline
- ~ Training Pipeline
- ~ Inferencing Pipeline
- ~ Retraining Pipeline
- ~ Training Pipeline Implementation
- ~ Introduction to Training Pipeline
- ~ Data Ingestion From Data Source
- ~ Data Validation
- ~ EDA, Data Preprocessing, Feature Engineering Model Selection
- ~ Model Training, Selection and Hyperparameter Tuning
- ~ Model Analysis and Evaluation
- ~ Model Push/ Export
- ~ Inference Pipeline Implementation
- ~ Introduction to Inference Pipeline
- ~ Understanding of the use of Artifact Generated by Training Pipeline
- ~ Data Validation
- ~ Data Preprocessing and Feature Engineering
- ~ Prediction using preprocessed data
- ~ Retraining Pipeline Implementation
- ~ Introduction to Retraining Pipeline
- ~ Model Analysis and Performance Monitoring of Prediction Pipeline
- ~ Creating Trigger to Initiate Model Retraining
- ~ Deployment of ML Pipeline on Cloud

- ~ *Schedule and Orchestrate Training Pipeline*
- ~ *Deployment of Inference Pipeline as an API*
- ~ *Deployment of Retraining Pipeline*
- ~ *Monitoring of System and Model Performance*
- ~ *Importance of Monitoring*
- ~ *Visualization of KPI and Other Indicator*
- ~ *System and Model Performance Visualization*
- ~ *Implementation of Alert and Notification to prevent Failure*
- ~ *Project Conclusion*

Yolo X and Yolo R

Topic Name : DATA SCIENCE

Sub-topic Name : COMPUTER VISION

Course link : <https://ineuron.ai/course/Yolo-X-and-Yolo-R>

Course Description :-

This course will help you to learn the practical implementations of YoloX & YoloR.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => YoloX
- => YoloR
- => Installation _ Setup
- => Get your custom data _ format it
- => Doing Annotation data
- => Handle your custom labels
- => Get pretrained weights

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Instructors :-

=> Boktiar Ahmed Bappy :

~ This is Bappy. I aim for simplicity in Data Science. Real Creativity won't make things more complex. Instead, I will simplify them, Interested in a Data Science Career and so develop myself accordingly. Data Scientist and lecturer with working experience in Machine Learning, Deep Learning, Microcontrollers and Electronics systems. Hands-on experience in classification, regression, clustering, computer vision, natural language processing and transfer learning models to solve challenging business problems. I have a huge interest in Robotics. I have innovated a lot of innovations, ideas, projects & robots and got a lot of achievements.

Curriculum details :-

=> YoloX :

- ~ Introduction to YOLOX
- ~ Installation _ Setup
- ~ Get your custom data _ format it
- ~ Doing Annotation data
- ~ Handle your custom labels
- ~ Get pretrained weights
- ~ Training YOLOX
- ~ Evaluation _ Visualize
- ~ Export Model _ Tasks

=> YoloR :

- ~ Introduction to YOLOR
- ~ Installation _ Setup
- ~ Custom Data
- ~ Data Annotation
- ~ Getting pretrained model
- ~ Model Training
- ~ Evaluation _ Saving models

Full Stack Data Science Feb'21 Batch

Topic Name : DATA SCIENCE

Sub-topic Name : FULL STACK DATA SCIENCE

Course link : <https://ineuron.ai/course/Full-Stack-Data-Science-Feb'21-Batch>

Course Description :-

This is a data science full stack live mentor led certification program along with full time one-year internship provided by iNeuron intelligence private limited, where you will learn all the stack required to work in data science, data analytics and big data industry including ML ops and cloud infrastructure and real time industry project and product development along with iNeuron product development team and you will contribute on various level with iNeuron .

Course Features :-

- => Full stack Data Science masters certification
- => Job guarantee otherwise refund
- => One year of internship
- => Online Instructor-led learning: Live teaching by instructors
- => 56 + hands-on industry real-time projects.
- => 400 hours live interactive classes.
- => Every week doubt clearing session after the live classes.
- => Lifetime Dashboard access.
- => Doubt clearing one to one
- => Doubt clearing through mail and skype support team
- => Assignment in all the module
- => Quiz in every module
- => A live project with real-time implementation
- => Resume building
- => Career guidance
- => Interview Preparation
- => Regular assessment

What you will learn :-

- => Python
- => Stats
- => Machine learning
- => Deep learning
- => Computer vision
- => Natural language processing
- => Data analytics
- => Big data
- => ML ops
- => Cloud
- => Data structure and algorithm
- => Architecture
- => Domain wise project
- => Databases
- => Negotiations skills
- => Mock interview
- => Interview preparation
- => Resume building after every module

Requirements :-

- => Dedication
- => Computer with i3 and above configuration

Instructors :-

=> Sunny Bhaveen Chandra :

~ Sr. Data Scientist and lecturer at iNeuron.ai with working experience in computer vision, natural language processing and embedded systems. Hands-on experience leveraging machine learning, deep learning, transfer learning models to solve challenging business problems. Also, he has a vast interest in Robotics.

=> krish naik :

~ Having 10+ years of experience in Data Science and Analytics with product architecture design and delivery. Worked in various product and service based Company. Having an experience of 5+ years in educating people and helping them to make a career transition.

=> Sudhanshu Kumar :

~ Having 8+ years of experience in Big data, Data Science and Analytics with product architecture design and delivery. Worked in various product and service based Company. Having an experience of 5+ years in educating people and helping them to make a career transition.

Curriculum details :-

=> Course introduction :

- ~ a. course overview and dashboard description
- ~ b. Introduction of data science and its application in day to day life
- ~ c. Programming language overview
- ~ d. Installation (tools: sublime, vscode, pycharm, anaconda, atom, jupyter notebook, kite)
- ~ e. Virtual environment
- ~ f. Why python

=> Python basic :

- ~ a. Introduction of python and comparison with other programming language
- ~ b. Installation of anaconda distribution and other python ide
- ~ c. Python objects, number & Booleans, strings.
- ~ d. Container objects, mutability of objects
- ~ e. Operators - arithmetic, bitwise, comparison and assignment operators, operators precedence and associativity
- ~ f. Conditions (if else, if-elif-else), loops (while, for)
- ~ g. Break and continue statement and range function

=> String objects :

- ~ a. basic data structure in python
- ~ b. String object basics
- ~ c. String inbuilt methods
- ~ d. Splitting and joining strings
- ~ e. String format functions

=> List object basics :

- ~ a. List methods
- ~ b. List as stack and queues
- ~ c. List comprehensions

=> Tuples, set, dictionaries & its function :

- ~ Dictionary object methods
- ~ Dictionary comprehensions
- ~ Dictionary view objects
- ~ Functions basics, parameter passing, iterators.
- ~ Generator functions
- ~ Lambda functions
- ~ Map, reduce, filter functions.

=> Memory management :

- ~ Multithreading
- ~ Multiprocessing

=> OOps concepts :

- ~ oops basic concepts.
- ~ Creating classes
- ~ Pillars of oops
- ~ Inheritance
- ~ Polymorphism
- ~ Encapsulation
- ~ Abstraction
- ~ Decorator
- ~ Class methods and static methods
- ~ Special (magic/dunder) methods
- ~ Property decorators - getters, setters, and deletes

=> Files :

- ~ Working with files
- ~ Reading and writing files
- ~ Buffered read and write
- ~ Other file methods.
- ~ Logging, debugger
- ~ Modules and import statements

=> Exception handling difference between exceptions and error :

- ~ Exceptions handling with try-except
- ~ Custom exception handling
- ~ List of general use exception
- ~ Best practice exception handling

=> Gui framework :

- ~ What is desktop and standalone application
- ~ Use of desktop app
- ~ Examples of desktop app
- ~ Tinker
- ~ Kivy

=> Database :

- ~ SQLite
- ~ MySQL
- ~ Mongo dB
- ~ NoSQL - Cassandra

=> Web API :

- ~ What is web API
- ~ Difference b/w API and web API
- ~ Rest and soap architecture
- ~ Restful services

=> Flask :

- ~ Flask introduction
- ~ Flask application
- ~ Open link flask
- ~ App routing flask
- ~ Url building flask
- ~ Http methods flask
- ~ Templates flask
- ~ Flask project: food app
- ~ Postman
- ~ Swagger

=> Django :

- ~ Django introduction
- ~ Django project: weather app
- ~ Django project: memes generator
- ~ Django project: blog app
- ~ Django project in cloud

=> Stream lit :

- ~ Stream lit introduction
- ~ Stream lit project structure
- ~ Stream lit project in cloud

=> Pandas basic :

- ~ Python pandas - series
- ~ Python pandas data frame
- ~ Python pandas panel
- ~ Python pandas - basic functionality
- ~ Reading data from different file system

=> Pandas advance :

- ~ Python pandas re indexing python
- ~ Pandas iteration
- ~ Python pandas sorting.
- ~ Working with text data options & customization
- ~ Indexing & selecting
- ~ Data statistical functions
- ~ Python pandas - window functions
- ~ Python pandas - date functionality
- ~ Python pandas time delta
- ~ Python pandas - categorical data
- ~ Python pandas visualization
- ~ Python pandas - iotools

=> Dask :

- ~ Dask Array
- ~ Dask Bag
- ~ Dask DataFrame
- ~ Dask Delayed
- ~ Dask Futures
- ~ Dask API
- ~ Dask SCHEDULING
- ~ Dask Understanding Performance
- ~ Dask Visualize task graphs
- ~ Dask Diagnostics (local)
- ~ Dask Diagnostics (distributed)
- ~ Dask Debugging
- ~ Dask Ordering

=> Python numpy :

- ~ Numpy - ND array object.
- ~ Numpy - data types.
- ~ Numpy - array attributes.
- ~ Numpy - array creation routines.
- ~ Numpy - array from existing.
- ~ Data array from numerical ranges.
- ~ Numpy - indexing & slicing.
- ~ Numpy advanced indexing.
- ~ Numpy broadcasting.
- ~ Numpy - iterating over array.
- ~ Numpy - array manipulation.
- ~ Numpy - binary operators.
- ~ Numpy - string functions.
- ~ Numpy - mathematical functions.
- ~ Numpy - arithmetic operations.
- ~ Numpy - statistical functions.
- ~ Sort, search & counting functions.

- ~ *Numpy - byte swapping.*
- ~ *Numpy - copies & views.*
- ~ *Numpy - matrix library.*
- ~ *Numpy - linear algebra*

=> Visualization :

- ~ *Matplotlib*
- ~ *Seaborn*
- ~ *Cufflinks*
- ~ *Plotly*
- ~ *Bokeh*

=> Statistics basic :

- ~ *Introduction to basic statistics terms*
- ~ *Types of statistics*
- ~ *Types of data*
- ~ *Levels of measurement*
- ~ *Measures of central tendency*
- ~ *Measures of dispersion*
- ~ *Random variables*
- ~ *Set*
- ~ *Skewness*
- ~ *Covariance and correlation*

=> Probability distribution function :

- ~ *Probability density/distribution function*
- ~ *Types of the probability distribution*
- ~ *Binomial distribution*
- ~ *Poisson distribution*
- ~ *Normal distribution (Gaussian distribution)*
- ~ *Probability density function and mass function*
- ~ *Cumulative density function*
- ~ *Examples of normal distribution*
- ~ *Bernoulli distribution*
- ~ *Uniform distribution*
- ~ *Z stats*
- ~ *Central limit theorem*
- ~ *Estimation*

=> Statistics advance :

- ~ *a Hypothesis*
- ~ *Hypothesis testings mechanism*
- ~ *P-value*
- ~ *T-stats*
- ~ *Student t distribution*
- ~ *T-stats vs. Z-stats: overview*
- ~ *When to use a t-tests vs. Z-tests*
- ~ *Type 1 & type 2 error*
- ~ *Bayes statistics (Bayes theorem)*
- ~ *Confidence interval(ci)*
- ~ *Confidence intervals and the margin of error*
- ~ *Interpreting confidence levels and confidence intervals*
- ~ *Chi-square test*
- ~ *Chi-square distribution using python*
- ~ *Chi-square for goodness of fit test*
- ~ *When to use which statistical distribution?*
- ~ *Analysis of variance (anova)*
- ~ *Assumptions to use anova*
- ~ *Anova three type*
- ~ *Partitioning of variance in the anova*
- ~ *Calculating using python*
- ~ *F-distribution*
- ~ *F-test (variance ratio test)*
- ~ *Determining the values of f*
- ~ *F distribution using python*

=> Linear algebra :

- ~ *linear algebra*
- ~ *Vector*
- ~ *Scaler*
- ~ *Matrix*
- ~ *Matrix operations and manipulations*
- ~ *Dot product of two vectors*
- ~ *Transpose of a matrix*
- ~ *Linear independence of vectors*
- ~ *Rank of a matrix*
- ~ *Identity matrix or operator*
- ~ *Determinant of a matrix*
- ~ *Inverse of a matrix*
- ~ *Norm of a vector*
- ~ *Eigenvalues and eigenvectors*
- ~ *Calculus*

=> Solving stats problem with python

=> Stats problem implementation with spicy

=> Introduction to machine learning :

- ~ *AI vs ml vs dl vs ds*
- ~ *Supervised, unsupervised, semi-supervised, reinforcement learning*
- ~ *Train, test, validation split*

- ~ Performance
- ~ Overfitting, under fitting
- ~ Bias vs variance

=> Feature engineering :

- ~ Handling missing data
- ~ Handling imbalanced data
- ~ Up-sampling
- ~ Down-sampling
- ~ Smote
- ~ Data interpolation
- ~ Handling outliers
- ~ Filter method
- ~ Wrapper method
- ~ Embedded methods
- ~ Feature scaling
- ~ Standardization
- ~ Mean normalization
- ~ Min-max scaling
- ~ Unit vector
- ~ Feature extraction
- ~ Pca (principle component analysis)
- ~ Data encoding
- ~ Nominal encoding
- ~ One hot encoding
- ~ One hot encoding with multiple categories
- ~ Mean encoding
- ~ Ordinal encoding
- ~ Label encoding
- ~ Target guided ordinal encoding
- ~ Covariance
- ~ Correlation check
- ~ Pearson correlation coefficient
- ~ Spearmans rank correlation
- ~ Vif

=> Feature selection :

- ~ Feature selection
- ~ Recursive feature elimination
- ~ Backward elimination
- ~ Forward elimination

=> Exploratory data analysis :

- ~ Feature engineering and selection.
- ~ Analyzing bike sharing trends.
- ~ Analyzing movie reviews sentiment.
- ~ Customer segmentation and effective cross selling.
- ~ Analyzing wine types and quality.
- ~ Analyzing music trends and recommendations.
- ~ Forecasting stock and commodity prices

=> Regression :

- ~ Linear regression
- ~ Gradient descent
- ~ Multiple linear regression
- ~ Polynomial regression
- ~ R square and adjusted r square
- ~ Rmse , mse, mae comparison
- ~ Regularized linear models
- ~ Ridge regression
- ~ Lasso regression
- ~ Elastic net
- ~ Complete end-to-end project with deployment on cloud and ui

=> Logistics regression :

- ~ Logistics regression in-depth intuition
- ~ In-depth mathematical intuition
- ~ In-depth geometrical intuition
- ~ Hyper parameter tuning
- ~ Grid search cv
- ~ Randomize search cv
- ~ Data leakage
- ~ Confusion matrix
- ~ Precision, recall, f1 score ,roc, auc
- ~ Best metric selection
- ~ Multiclass classification in lr
- ~ Complete end-to-end project with deployment in multi cloud platform

=> Decision tree :

- ~ Decision tree classifier
- ~ In-depth mathematical intuition
- ~ In-depth geometrical intuition
- ~ Confusion matrix
- ~ Precision, recall, f1 score ,roc, auc
- ~ Best metric selection
- ~ Decision tree repressor
- ~ In-depth mathematical intuition
- ~ In-depth geometrical intuition
- ~ Performance metrics
- ~ Complete end-to-end project with deployment in multi cloud platform

=> Support vector machines :

- ~ Linear svm classification
- ~ In-depth mathematical intuition
- ~ In-depth geometrical intuition
- ~ Soft margin classification
- ~ Nonlinear svm classification
- ~ Polynomial kernel
- ~ Gaussian, rbf kernel
- ~ Data leakage
- ~ Confusion matrix
- ~ precision, recall, f1 score, roc, auc
- ~ Best metric selection
- ~ Svm regression
- ~ In-depth mathematical intuition
- ~ In-depth geometrical intuition
- ~ Complete end-to-end project with deployment

=> Nave Bayes :

- ~ Bayes theorem
- ~ Multinomial nave Bayes
- ~ Gaussian nave Bayes
- ~ Various type of Bayes theorem and its intuition
- ~ Confusion matrix
- ~ precision, recall, f1 score, roc, auc
- ~ Best metric selection
- ~ Complete end-to-end project with deployment

=> Ensemble techniques and its types :

- ~ Definition of ensemble techniques
- ~ Bagging technique
- ~ Bootstrap aggregation
- ~ Random forest (bagging technique)
- ~ Random forest repressor
- ~ Random forest classifier
- ~ Complete end-to-end project with deployment

=> Boosting :

- ~ Boosting technique
- ~ Ada boost
- ~ Gradient boost
- ~ Xgboost
- ~ Complete end-to-end project with deployment

=> Stacking :

- ~ Stacking technique
- ~ Complete end-to-end project with deployment

=> Knn :

- ~ Knn classifier
- ~ Knn repressor
- ~ Variants of knn
- ~ Brute force knn
- ~ K-dimension tree
- ~ Ball tree
- ~ Complete end-to-end project with deployment

=> Dimensionality reduction :

- ~ The curse of dimensionality
- ~ Dimensionality reduction technique
- ~ Pca (principle component analysis)
- ~ Mathematics behind pca
- ~ Scree plots
- ~ Eigen-decomposition approach

=> Clustering :

- ~ Clustering and their types
- ~ K-means clustering
- ~ K-means++
- ~ Batch k-means
- ~ Hierarchical clustering
- ~ Dbscan
- ~ Evaluation of clustering
- ~ Homogeneity, completeness and v-measure
- ~ Silhouette coefficient
- ~ Davies-bouldin index
- ~ Contingency matrix
- ~ Pair confusion matrix
- ~ Extrinsic measure
- ~ Intrinsic measure
- ~ Complete end-to-end project with deployment

=> Anomaly detection :

- ~ Anomaly detection types
- ~ Anomaly detection applications
- ~ Isolation forest anomaly detection algorithm
- ~ Density-based anomaly detection (local outlier factor) algorithm
- ~ Support vector machine anomaly detection algorithm
- ~ Dbscan algorithm for anomaly detection
- ~ Complete end-to-end project with deployment

=> Time-series :

- ~ What is a time series?
- ~ Old techniques
- ~ Arima
- ~ Acf and pacf
- ~ Time-dependent seasonal components.
- ~ Autoregressive (ar),
- ~ Moving average (ma) and mixed arma- modeler.
- ~ The random walk model.
- ~ Box-jenkins methodology.
- ~ Forecasts with arima and var models.
- ~ Dynamic models with time-shifted explanatory variables.
- ~ The koyck transformation.
- ~ Partial adjustment and adaptive expectation models.
- ~ Granger's causality tests.
- ~ Stationarity, unit roots and integration
- ~ Time series model performance
- ~ Various approach to solve time series problem
- ~ Complete end-to-end project with deployment
- ~ Prediction of nifty stock price and deployment

=> NLP basic :

- ~ Tokenization
- ~ Pos tags and chunking
- ~ Stop words
- ~ Stemming and lemmatization
- ~ Named entity recognition (ner)
- ~ Word vectorization (word embedding)
- ~ Tfidf
- ~ Complete end-to-end project with deployment

=> Machine learning pipeline :

- ~ Aws segmaker
- ~ Aure ml studio
- ~ Ml flow
- ~ Kube flow

=> Model retraining approach

=> Auto ML :

- ~ H2o
- ~ Pycaret
- ~ Auto sklearn
- ~ Auto time series
- ~ Auto viml
- ~ Auto gluon
- ~ Auto viz
- ~ Tpot
- ~ Auto neuro

=> Neural network a simple perception. :

- ~ Detail mathematical explanation
- ~ Neural network overview and its use case.
- ~ Various neural network architect overview.
- ~ Use case of neural network in nlp and computer vision.
- ~ Activation function -all name
- ~ Multilayer network.
- ~ Loss functions. - all 10
- ~ The learning mechanism.
- ~ Optimizers. - all 10
- ~ Forward and backward propagation.
- ~ Weight initialization technique
- ~ Vanishing gradient problem
- ~ Exploding gradient problem
- ~ Visualization of nn

=> Hardware setup GPU :

- ~ Gpu introduction.
- ~ Various type of gpu configuration.
- ~ Gpu provider and its pricing.
- ~ Paper space gpu setup.
- ~ Running model in gpu

=> Tensor flow installation environment setup for deep learning :

- ~ Colab pro setup
- ~ Tensor flow installation 2.0 .
- ~ Tensor flow installation 1.6 with virtual environment.
- ~ Tensor flow 2.0 function.
- ~ Tensor flow 2.0 neural network creation.
- ~ Tensor flow 1.6 functions.
- ~ Tensor flow 1.6 neural network and its functions.
- ~ Keras introduction.
- ~ Keras in-depth with neural network creation.
- ~ Mini project in tensorflow.
- ~ Tensorspace
- ~ Tensorboard integration
- ~ Tensorflow playground
- ~ Netron

=> Pytorch :

- ~ pytorch installation.
- ~ Pytorch functional overview.

~ *Pytorch neural network creation.*

=> **Mxnet :**

~ *Mxnet installation*
~ *Mxnet in depth function overview*
~ *Mxnet model creation and training*

=> **Keras tuner :**

~ *Keras tuner installation and overview*
~ *Finding best parameter from keras tuner*
~ *Keras tuner application across various neural network*

=> **Cnn overview :**

~ *Cnn definition*
~ *Various cnn based architecture*
~ *Explanation end to end cnn network*
~ *Cnn explainer*
~ *Training cnn*
~ *Deployment in azure cloud*
~ *Performance tuning of cnn network*

=> **Advance computer vision part 1 :**

~ *Various cnn architecture with research paper and mathematics*
~ *Lenet-5 variants with research paper and practical*
~ *Alexnet variants with research paper and practical*
~ *Googlenet variants with research paper and practical*
~ *Transfer learning*
~ *Vggnet variants with research paper and practical*
~ *Resnet variants with research paper and practical*
~ *Inception net variants with research paper and practical*
~ *Darknet variants with research paper and practical*

=> **Advance computer vision part 2 :**

~ *Object detection in-depth*
~ *Transfer learning*
~ *Rcnn with research paper and practical*
~ *Fast rcnn with research paper and practical*
~ *Faster r cnn with research paper and practical*
~ *Ssd with research paper and practical*
~ *Ssd lite with research paper and practical*

=> **Training of custom object detection :**

~ *Tfod introduction*
~ *Environment setup with tfod*
~ *Gpu vs tpu vs cpu*
~ *Various gpu comparison*

=> **Advance computer vision part 3 :**

~ *Yolo v1 with research paper and practical*
~ *Yolo v2 with research paper and practical*
~ *Yolo v3 with research paper and practical*
~ *Yolo v4 with research paper and practical*
~ *Yolo v5 with research paper and practical*
~ *Retina net*
~ *Face net*
~ *Detectron2 with practical and live testing*

=> **Object segmentation :**

~ *Semantic segmentation*
~ *Panoptic segmentation*
~ *Masked rcnn*
~ *Practical with detectron*
~ *Practical with tfod*

=> **Object tracking :**

~ *Detail of object tracking*
~ *Kalman filtering*
~ *Sort*
~ *Deep sort*
~ *Object tracking live project with live camera testing*

=> **OCR :**

~ *Introduction to ocr*
~ *Various framework and api for ocr*
~ *Practical implementation of ocr*

=> **Advance NLP with deep-learning :**

~ *Overview computational linguistic.*
~ *History of nlp.*
~ *Why nlp*
~ *Use of nlp*

=> **Text processing importing text. :**

~ *Web scrapping.*
~ *Text processing*
~ *Understanding regex.*
~ *Text normalization*
~ *Word count.*
~ *Frequency distribution.*
~ *Text annotation.*
~ *Use of annotator.*
~ *String tokenization*

- ~ *Annotator creation.*
- ~ *Sentence processing.*
- ~ *Lemmatization in text processing*
- ~ *Pos.*
- ~ *Named entity recognition*
- ~ *Dependency parsing in text.*
- ~ *Sentimental analysis*

=> **Spacy :**

- ~ *Spacy overview.*
- ~ *Spacy function*
- ~ *Spacy function implementation in text processing.*
- ~ *Pos tagging, challenges and accuracy.*
- ~ *Entities and named entry recognition*
- ~ *Interpolation, language models*
- ~ *Nltk*
- ~ *Text blob*
- ~ *Stanford nlp*

=> **RNN :**

- ~ *Recurrent neural networks.*
- ~ *Long short term memory (lstm)*
- ~ *Bi lstm.*
- ~ *Stacked lstm*
- ~ *Gru implementation.*
- ~ *Building a story writer using character level rnn.*

=> **Word embedding :**

- ~ *Word embedding*
- ~ *Co-occurrence vectors*
- ~ *Word2vec*
- ~ *Doc2vec*

=> **Attention based model :**

- ~ *Seq 2 seq.*
- ~ *Encoders and decoders.*
- ~ *Attention mechanism.*
- ~ *Attention neural networks*
- ~ *Self-attention*

=> **Transfer learning in nlp :**

- ~ *Introduction to transformers.*
- ~ *Bert model.*
- ~ *Elmo model.*
- ~ *Gpt1 model*
- ~ *Gpt2 model.*
- ~ *Albert model.*
- ~ *Distilbert model*

=> **Deployment of model and performance tuning :**

- ~ *Deep learning model deployment strategies.*
- ~ *Deep learning project architecture*
- ~ *Deep learning model deployment phase.*
- ~ *Deep learning model retraining phase.*
- ~ *Deep learning model deployment in aws.*
- ~ *Deep learning model deployment in azure.*
- ~ *Deep learning model deployment in gcloud.*

=> **Big data introduction :**

- ~ *What is big data?*
- ~ *Big data application*
- ~ *Big data pipeline*

=> **Hadoop :**

- ~ *Hadoop introduction*
- ~ *Hadoop setup and installation*

=> **Spark :**

- ~ *Spark*
- ~ *Spark overview.*
- ~ *Spark installation.*
- ~ *Spark rdd.*
- ~ *Spark data frame.*
- ~ *Spark architecture.*
- ~ *Spark ml lib*
- ~ *Spark NLP*
- ~ *Spark linear regression*
- ~ *Spark logistic regression*
- ~ *Spark decision tree*
- ~ *Spark naive bayes*
- ~ *Spark xg boost.*
- ~ *Spark time series*
- ~ *Spark deployment in local server*
- ~ *Spark job automation with*
- ~ *Scheduler*

=> **Kafka :**

- ~ *Kafka introduction*
- ~ *Kafka installation*
- ~ *Spark streaming*
- ~ *Spark with Kafka*

=> Tableau :

- ~ Talking about Business Intelligence
- ~ Tools and Methodologies used in BI
- ~ Why Visualization is getting more popular
- ~ Why Tableau?
- ~ Gartner Magic Quadrant of Market Leaders
- ~ Future business impact of BI
- ~ Tableau Products
- ~ Tableau Architecture
- ~ BI Project Execution
- ~ Tableau Installation in local system
- ~ Introduction to Tableau Prep
- ~ Tableau Prep Builder User Interface
- ~ Data Preparation techniques using Tableau Prep Builder tool
- ~ How to connect Tableau with different data source
- ~ Visual Segments
- ~ Visual Analytics in depth
- ~ Filters, Parameters & Sets
- ~ Tableau Calculations using functions
- ~ Tableau Joins
- ~ Working with multiple data source (Data Blending)
- ~ Building Predictive Models
- ~ Dynamic Dashboards and Stories
- ~ Sharing your Reports
- ~ Tableau Server
- ~ User Security
- ~ Scheduling

=> Power BI :

- ~ Power BI introduction and overview
- ~ Key Benefits of Power BI
- ~ Power BI Architecture
- ~ Power BI Process
- ~ Components of Power BI
- ~ Power BI - Building Blocks
- ~ Power BI vs other BI tools
- ~ Power Installation
- ~ Overview of Power BI Desktop
- ~ Data Sources in Power BI Desktop
- ~ Connecting to a data Sources
- ~ Query Editor in Power BI
- ~ Views in Power BI
- ~ Field Pane
- ~ Visual Pane
- ~ Custom Visual Option
- ~ Filters
- ~ Introduction to using Excel data in Power BI
- ~ Exploring live connections to data with Power BI
- ~ Connecting directly to SQL Azure, HD Spark, SQL Server Analysis Services/ My SQL
- ~ Import Power View and Power Pivot to Power BI
- ~ Power BI Publisher for Excel
- ~ Content packs
- ~ Introducing Power BI Mobile
- ~ Power Query Introduction
- ~ Query Editor Interface
- ~ Clean and Transform your data with Query Editor
- ~ Data Type
- ~ Column Transformations vs Adding Columns
- ~ Text Transformations
- ~ Cleaning irregularly formatted data -Transpose
- ~ Date and Time Calculations
- ~ Advance editor: Use Case
- ~ Query Level Parameters
- ~ Combining Data Merging and Appending
- ~ Data Modelling
- ~ Calculated Columns
- ~ Measures/New Quick Measures
- ~ Calculated Tables
- ~ Optimizing Data Models
- ~ Row Context vs Set Context
- ~ Cross Filter Direction
- ~ Manage Data Relationship
- ~ Why is DAX important?
- ~ Advanced calculations using Calculate functions
- ~ DAX queries

=> Reinforcement Learning

Project details :-

=> Python Project :

- ~ Weeding script
- ~ Image resizing
- ~ Jupyter notebook merging, reading etc.
- ~ Sending emails
- ~ Weather app
- ~ Memes generator
- ~ Food log app
- ~ Web scrapping

- ~ Web crawlers for image data sentiment analysis and product review sentiment analysis.
- ~ Integration with web portal.
- ~ Integration with rest api, web portal and mongo db. on azure
- ~ Deployment on web portal on azure.
- ~ Text mining
- ~ Social media data churn
- ~ Mass copy, paste

=> Chatbot projects :

- ~ Chatbot using Microsoft Luis
- ~ Chatbot using google dialog flow
- ~ Chatbot using amazon lex
- ~ Chatbot using rasa nlu
- ~ Deployment of Chabot with web , telegram , WhatsApp, skype

=> Major projects :

- ~ Healthcare analytics prediction of medicines based on Fitbit band.
- ~ Revenue forecasting for startups.
- ~ Prediction of order cancellation at the time of ordering inventories.
- ~ anomaly detection in inventory packaged material.
- ~ Fault detection in wafers based on sensor data.
- ~ Demand forecasting for fmcg product.
- ~ Threat identification in security system.
- ~ Defect detection in vehicle engine.
- ~ Food price forecasting with zomato dataset.
- ~ Fault detection in wafers based on sensor data.
- ~ Cement strength reg.
- ~ Credit card fraud.
- ~ Forest cover classification.
- ~ Fraud detection.
- ~ Income prediction.
- ~ Mushroom classifier.
- ~ phishing classifier
- ~ Thyroid detection.
- ~ Visibility climate

=> Computer vision project :

- ~ Traffic surveillance system.
- ~ Object identification.
- ~ Object tracking.
- ~ Object classification.
- ~ Tensorflow object detection.
- ~ Image to text processing.
- ~ Speech to speech analysis.
- ~ Vision based attendance system

=> Mini NLP project :

- ~ Machine translation.
- ~ Abstractive text summarization.
- ~ Keyword spotting.
- ~ Language modelling.
- ~ Document summarization

=> Nlp transfer learning project :

- ~ Deployment and integration with UI machine translation.
- ~ Question answering (like chat bot)
- ~ Sentiment analysis imdb.
- ~ Text search (with synonyms).
- ~ Text classifications.
- ~ Spelling corrector.
- ~ Entity (person, place or brand) recognition.
- ~ Text summarization.
- ~ Text similarity (paraphrase).
- ~ Topic detection.
- ~ Language identification.
- ~ Document ranking.
- ~ Fake news detection
- ~ Plagiarism checker
- ~ Text summarization extractive
- ~ Text summarization abstractive.

=> NLP end to end project with architecture and deployment :

- ~ Movie review using bert
- ~ Ner using bert
- ~ Pos bert
- ~ Text generation gpt 2
- ~ Text summarization xlnet
- ~ Abstract bert
- ~ Machine translation
- ~ Nlp text summarization custom
- ~ Keras/tensorflow
- ~ Language identification
- ~ Text classification using fast bert
- ~ Neuralcore
- ~ Detecting fake text using gltr with bert and gpt2
- ~ Fake news detector using gpt2
- ~ Python plagiarism checker type a message
- ~ Question answering

Dask

Topic Name : DATA SCIENCE

Sub-topic Name : MACHINE LEARNING

Course link : <https://ineuron.ai/course/Dask>

Course Description :-

Dask is a flexible library for parallel computing in Python. It can easily handle large data which enables users to perform ml related tasks at scale.

Course Features :-

- => Self-Paced Classes
- => Real-time Project
- => Assignment in all modules
- => Quiz in every module
- => Completion Certificate

What you will learn :-

- => Dask Arrays
- => Dask Dataframes
- => Dask Bags
- => ML with Dask

Requirements :-

- => Little bit of Python Knowledge
- => Dedication
- => Internet Connection

Instructors :-

- => MD Imran :
 - ~ Working as Data Scientist with experience in solving real world business problems across different domains.

Curriculum details :-

- => Introduction :
 - ~ The course Overview Preview
 - ~ Introduction to Dask Preview
 - ~ Dask Alternatives
 - ~ Advantages of using dask
 - ~ Limitations of task
 - ~ Dask Setup Preview
- => Understanding dask arrays :
 - ~ Introduction to blocked algorithms
 - ~ Hands on with DASK Arrays
 - ~ Digging deeper into dask arrays
 - ~ performance comparison with numpy arrays Preview
 - ~ creating universal numpy functions with dask
 - ~ Limitations of Dask
- => Parallelizing python code with DASK :
 - ~ Lazy Evaluation
 - ~ using `dask.delayed`
 - ~ understand task graphs
- => Understanding Dask Dataframes :
 - ~ Introduction to dask dataframes
 - ~ exploring dask dataframes
 - ~ creating dask dataframes
 - ~ loading large datasets with dask dataframes
 - ~ analyzing data with dask dataframes
 - ~ limitations of dask dataframes
- => Exploring Dask Bags :
 - ~ Introduction to dask bags
 - ~ creating and storing dask bags
 - ~ manipulating dask bags
 - ~ word count example using dask bags
 - ~ Limitations of Dask Bags
- => Distributed computing with dask :
 - ~ overview of distributed computing with dask
 - ~ setting up your dask cluster
 - ~ understanding dask schedulers

~ Exploring dask dashboard UI

=> Machine Learning with Dask :

~ Introduction to dask ML Preview

~ using dask ML for regression

~ using dask ML for Classification

The Ultimate Guide To OpenAI GPT-3 & Fine Tune with Custom Data

Topic Name : DATA SCIENCE

Sub-topic Name : NLP PROJECT

Course link : <https://ineuron.ai/course/The-Ultimate-Guide-To-OpenAI-GPT-3-&-Fine-Tune-with-Custom-Data>

Course Description :-

Generative Pre-trained Transformer 3 (GPT-3; stylized GPT3) is an autoregressive language model that uses deep learning to produce human-like text. Given an initial text as prompt, it will produce text that continues the prompt. The architecture is a standard transformer network (with a few engineering tweaks) with the unprecedented size of 2048-token-long context and 175 billion parameters (requiring 800 GB of storage). The training method is "generative pretraining", meaning that it is trained to predict what the next token is. The model demonstrated strong few-shot learning on many text-based tasks.

Course Features :-

- => Do Everything In Industry Grade Lab
- => Learn As Per Your Timeline
- => Hands-On Industry Real-Time Projects.
- => Self Paced Learning
- => Dashboard Access
- => Course Materials
- => Assignments

What you will learn :-

- => Real Time Projects
- => OpenAI GPT-3 & Fine Tune with Custom data
- => What is GPT-3?
- => Demo: Build a paraphraser, Chatbot, Summarization
- => OpenAI playground to develop prompts.
- => Understand various engines and GPT-3 prompt parameters.
- => Create novel datasets with GPT-3 and Streamlit UI.
- => Zero-shot and few-shot prompts.
- => Fine tuning GPT3

Requirements :-

- => System with minimum i3 processor or better
- => At least 4 GB of RAM
- => Working internet connection
- => Dedication to learn

Instructors :-

=> Boktiar Ahmed Bappy :

~ This is Bappy. I aim for simplicity in Data Science. Real Creativity won't make things more complex. Instead, I will simplify them, Interested in a Data Science Career and so develop myself accordingly. Data Scientist and lecturer with working experience in Machine Learning, Deep Learning, Microcontrollers and Electronics systems. Hands-on experience in classification, regression, clustering, computer vision, natural language processing and transfer learning models to solve challenging business problems. I have a huge interest in Robotics. I have innovated a lot of innovations, ideas, projects & robots and got a lot of achievements.

Curriculum details :-

- => Welcome to the Course :
 - ~ Course Overview
 - ~ Dashboard Introduction
- => Project :- The Ultimate Guide To OpenAI GPT-3 & Fine Tune with Custom Data :
 - ~ What is GPT-3?
 - ~ Demo: Build a paraphraser, Chatbot, Summarization
 - ~ OpenAI playground to develop prompts.
 - ~ Understand various engines/models and GPT-3 prompt parameters.
 - ~ Create novel datasets with GPT-3 and Streamlit UI.
 - ~ Zero-shot and few-shot prompts.
 - ~ Fine tuning GPT3
 - ~ Conclude the project
 - ~ Assignments & External Resources

AIOPS Course

Topic Name : DATA SCIENCE

Sub-topic Name : MLOPS

Course link : <https://ineuron.ai/course/AIOPS-Course>

Course Description :-

These days the most in-demand technical expertise is Artificial Intelligence Operations (AIOps). It aids in the application of DevOps principles to AI product development. This course will cover multiple ways to implement AIOps methodology in ML and DL projects, including implementation on various clouds such as AWS, Azure, GCP, etc.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => AI/MLOps
- => Mlflow
- => Linux
- => GIT
- => DVC
- => Docker
- => Kubernetes
- => TFX

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Instructors :-

=> Sunny Bhavleen Chandra :

~ Sr. Data Scientist and lecturer at iNeuron.ai with working experience in computer vision, natural language processing and embedded systems. Hands-on experience leveraging machine learning, deep learning, transfer learning models to solve challenging business problems. Also, he has a vast interest in Robotics.

=> Sourangshu Pal :

~ Visual Computing Engineer and instructor at iNeuron.ai having 3 years of diverse experience in the discipline of visual computing with specialization in Deep Learning and Computer Graphics. Loves to analyze, process, and model visual data then interpret the insights to create actionable plans for solving challenging business problems.

=> Ritesh Yadav :

~ Ritesh is truly passionate about data science, machine learning and DevOps in general, he likes what he does, and is keen to learn. Currently, He is working as a Jr. Data Scientist at iNeuron.ai. He also loves to contribute to Open Source Projects, which are mainly under CNCF Landscape. Ritesh loves to work in Cloud-Native technologies and Golang (Go). Apart from this, Ritesh has been actively involved in the open-source community for over a year, helping many open-source DevOps tools and CNCF Projects like Porter, Meshery, Keptn, TensorFlow, and Thanos through his contributions.

Curriculum details :-

=> AIOPS Course :

- ~ 17th July induction session
- ~ 18th July Live Class Introduction to AI/MLOps
- ~ 20th July Live doubt clearing Session
- ~ 24th July Live Class MLOps level1 workflow
- ~ 25th July Live Class MLOps level2 workflow
- ~ 31st July Live Class Linux Setup in Virtual Box
- ~ 1st August Live Class SSH Setup to Virtual Box
- ~ 3rd August doubt clearing session
- ~ 4th August live class General Discussion
- ~ 7th August live class Linux Introduction Evolution of Linux Differences b/w Windows Linux Monolithic Kernel
- ~ 8th August Live Class Kernel Boot Process Linux Runlevels Filetypes in Linux Package Management
- ~ 10th August Doubt Clearing session
- ~ 14th August Live Class Components in Linux OS File system Linux Commands

- ~ 17th August Doubt Clearing Session
- ~ 21st August Live Class Archiving Compression VI Editor Runlevel changes Accounts User Mangement
- ~ 22nd August live class Searching in LinuxGrep Vi Editor File Permissions
- ~ 24th August Doubt clearing session
- ~ 28th August Live Class Scp Passwordless SSH Cronjobs Service Mangement
- ~ 30th August Live class Linux in AWS Part1
- ~ 31st August Doubt Clearing session
- ~ 4th September Live Class Linux in AWS Part2
- ~ 5th September live Class GIT Introduction
- ~ 7th September doubt clearing session
- ~ 11th September Live class GIT-Working with Repos History Check
- ~ 12th September Live Class Merging Merge Conflicts
- ~ 14th September Doubt Clearing session
- ~ 18th September live class Rebasing Interactive Rebasing Amend
- ~ 19th September Live class Revert Reset Cherry Pick Reflog Checkout
- ~ 21st September Doubt Clearing session
- ~ 25th September Live Class DVC introduction
- ~ 26th September Live Class DVC basic ML use case
- ~ 28th September Doubt clearing session
- ~ 3rd October Live Class DVC basic ML use case
- ~ 5th October Doubt Clearing session
- ~ 9th October Live class DVC DL use case with tensorflow
- ~ 10th October Live Class DVC basics DL-Tf use case continued
- ~ 12th October Doubt clearing session
- ~ 23rd October Live Class DVC basics DL-Tf usecase till stage 04
- ~ 25th October Live Class DVC basics DL Tf - final stage COMPLETED
- ~ 26th October Doubt Clearing session
- ~ 30th October Live Class DVC -NLP use case
- ~ 31st October Live Class DVC -NLP usecase completed till plots
- ~ 2nd November Live Class EXTRA SESSION: CML
- ~ 7th November Live Class Docker Installation
- ~ 13th November Live Class Docker Session -1
- ~ 14th November Live Class Docker Session -2
- ~ 16th November Doubt Clearing session
- ~ 20th November Live Class Docker Project
- ~ 21st November Live Class Docker Project
- ~ 23rd November Doubt clearing session
- ~ 27th November Live Class Bank-Note Authentication Docker project
- ~ 28th November Live Class Docker Task
- ~ 4th December Live Class Docker with Gpu Nlp Application Deployment
- ~ 5th December Live Class Docker Compose
- ~ 18th Dec Live Class Recording Microservice vs Monolithic
- ~ 19th Dec Live Class Recording Microservice vs Monolithic and Docker Networking
- ~ 08th Jan Live Class Recordings Kubernetes session 1 Introduction to Kubernetes
- ~ 9th Jan Live Class Kubernetes session 2: Installation and basics
- ~ 15th Jan Live Class Kubernetes session 3: Basic Commands
- ~ 22 Jan Live Class
- ~ 23rd Jan Live Class
- ~ Jan 29th Live Class
- ~ Jan 31th Jan
- ~ 6th Feb Live Class
- ~ Feb 12th Live Class
- ~ 8th Feb live Class
- ~ 13th Feb Live Class MLFLOW session 01
- ~ 19th Feb Live Class
- ~ 20th Feb Live Class Part1 MLFLOW session 02
- ~ 20th Feb Live Class Part2 MLFLOW session 02
- ~ 26th Feb live class MLFLOW session 03
- ~ 27th Feb Live Class MLFLOW session 04
- ~ March 5th Live Class MLFLOW using AWS Infrastructure
- ~ March 6th Live Class MLFLOW using AWS Infrastructure - 2
- ~ 12th March Live Class MLflow using Azure Infrastructure -1
- ~ 13th march Live Class MLflow using Azure Infrastructure -2
- ~ 20th March Live Class MLOps using GCP Infrastructure with Terraform automation
- ~ 26th March Live Class Kubeflow - 1
- ~ 27th March Live class Kubeflow - 2
- ~ 2nd April Live Class Kubeflow - 3
- ~ 3rd April Live Class Part 1 Introduction of TFX based ML Pipeline
- ~ 3rd April Live Class Part 2 Introduction of TFX based ML Pipeline
- ~ 9th April Live Class TFX project initial setup
- ~ 10th April Live Class TFX data ingestion custom component
- ~ 16th April Live Class Data Validation and Feature Engineering using TFX
- ~ 17th April Live Class Model training and Model Evaluation using TFX
- ~ 23rd April Live Class Deployment of source code at s3 bucket and Apache Airflow
- ~ 24th April Live Class Publishing python projects as python package at PYPI
- ~ 30th April Live Class Digital Ocean CI CD (Text Classification)
- ~ 1st May Live Class LSTM TEXT CLASSIFICATION
- ~ 7th May Live Class End to End Aiops project with infrastructure and endpoint monitoring
- ~ 8th May Live Class End to End Aiops project with infrastructure and endpoint monitoring

Drone Live Class

Topic Name : K12

Sub-topic Name : CLASS10

Course link : <https://ineuron.ai/course/Drone-Live-Class>

Course Description :-

iNeuron has created a course in collaboration with industry experts who are achieving incredible things with autonomous robotics. You can learn about the process of making a drone or how drones are used in various sectors. The majority of drones are ready to go to the skies. We provide Robotics concepts to teach you about dynamics and control so you can get a better understanding of how that works. We present an outline of Drones and Autonomous Systems for individuals who want to learn more about their mechanics. Each lecture is intended to lay the groundwork for how autonomous systems can alter our perceptions of robotics.

Course Features :-

- => Online live classes
- => Doubt Clearing
- => Live-Class Recording
- => Real-time Project
- => Assignment in all modules
- => Quiz in every module
- => Career Counselling
- => Completion Certificate

What you will learn :-

- => Basic understanding of UAV and Maneuvering of Aircraft.
- => Construction(Multirotor drone basic)
- => Drone Body Assembly
- => Transmitter and Receiver
- => Flight controller
- => Calibrations

Requirements :-

- => No prior experience of Drones

Instructors :-

- => Sunny Bhavleen Chandra :

~ Sr. Data Scientist and lecturer at iNeuron.ai with working experience in computer vision, natural language processing and embedded systems. Hands-on experience leveraging machine learning, deep learning, transfer learning models to solve challenging business problems. Also, he has a vast interest in Robotics.

Curriculum details :-

- => Introduction :

- ~ What is this course all about?
- ~ What is in this course?
- ~ Pre-knowledge required.

- => Basic understanding of UAV and Maneuvering of Aircraft. :

- ~ Aerodynamics of UAV
- ~ Forces of Flight
- ~ Theory of flight
- ~ Centre of Gravity
- ~ Thrust to Weight Ratio
- ~ Mach Number.

- => Types of UAV

<Parts of UAV and their diagrams > :

- ~ Multi-Rotor
- ~ Fixed wings
- ~ Single rotor
- ~ Hybrid VTOL
- ~ Their Pros, and Cons

- => Construction(Multirotor drone basic) :

- ~ Axis of rotation for Drone
- ~ Components needed to construct a drone.
- ~ Terminologies
- ~ Tools needed for drone construction
- ~ Conclusion.

=> Explaining each component and its functions(Motors, Frame, and ESC) :

- ~ Different types of drone frame
- ~ How to choose it
- ~ Motors: what are BLDC motors
- ~ Why we are using it.
- ~ Their Rating and Explaining how to correctly select the motors for the drone.

=> Explaining each component and its functions(Power Distribution Boards & Common Module wire and ESC) :

- ~ How to solder ESC power wires
- ~ How to solder the Battery power wires and connectors

=> Explaining each component and its functions(Drone Body Assembly) :

- ~ Attaching the arms to the frame of the drone.
- ~ Attaching the Motors to the Arms
- ~ Attaching the ESC to the arm and connecting them to the motors.
- ~ Propellers and how to select a propeller for your drone, and how to distinguish between clockwise and anti-clock propellers

=> Explaining each component and its functions(Transmitter and Receiver) :

- ~ Introduction about the Transmitter and Receiver
- ~ Binding the Receiver with the Transmitter
- ~ Explaining about Telemetry
- ~ Explaining the features Of the Transmitter with the help of servo and how we can utilize all of its features

=> Explaining each component and its functions(Flight controller) :

- ~ Explaining Different aspects of Pixhawk flight controller
- ~ Connecting the GPS module to the flight controller
- ~ Attaching all the necessary components to the flight controller (Switch Buzzer, Telemetry, Power module)
- ~ Connecting ESC to the Pixhawk with the correct order
- ~ Connecting RC receiver with the Pixhawk

=> Explaining each component and its functions(Battery and Charge) :

- ~ Explain the Battery and charger

=> Explaining each component and its functions(Software MISSION PLANNER) :

- ~ 1. Downloading and setting up the Mission Planner
- ~ 2. Connecting Pixhawk with the Mission Planner
- ~ 3. Explaining various features of Mission Planner

=> Explaining each component and its functions(Calibrations Part) :

- ~ Calibrating GPS and other onboard sensors with the Pixhawk using Mission Planner
- ~ Calibrating ESC and Motors

=> Explaining each component and its functions(FIRST FLIGHT) :

- ~ Taking First Flight
- ~ Demonstrating how to control UAV using the remote controller

Language Identification

Topic Name : DATA SCIENCE

Sub-topic Name : NLP PROJECT

Course link : <https://ineuron.ai/course/Language-Identification>

Course Description :-

This is an audio classification project in which we will use Pytorch for audio processing and CNN for audio classification. We will use Indian language audio data from four classes, Hindi, Tamil, Telugu, and Kannada, and predict the language spoken in the audio.

Course Features :-

- => Do Everything In Industry Grade Lab
- => Learn As Per Your Timeline
- => Hands-On Industry Real-Time Projects.
- => Self Paced Learning
- => Dashboard Access

What you will learn :-

- => Real Time Projects
- => Language Identification
- => Audio preprocessing steps to build
- => Train and evaluate Deep learning models in PyTorch
- => Creating custom PyTorch dataset and dataloader
- => Use convolution neural network for audio classification.
- => Modular coding approach for training and prediction pipeline
- => Building Flask App
- => Learn about GCP basics
- => CICD tool like Github Actions for deployment

Requirements :-

- => System with minimum i3 processor or better
- => At least 4 GB of RAM
- => Working internet connection
- => Dedication to learn

Instructors :-

=> Aravind S :

~ Data scientist with over a year of experience in developing advanced deep learning projects, core expertise in machine learning and NLP, proficient in data preprocessing and model building, and has closely mentored over 100 students from various domains.

Curriculum details :-

=> Welcome to the Course :

- ~ Course Overview
- ~ Dashboard Introduction

=> Project :- Language Identification :

- ~ Introduction of Instructor
- ~ Project Overview
- ~ End Notes
- ~ Problem Description
- ~ Understand the application scope
- ~ End Notes
- ~ Solution Description
- ~ Notebook Walkthrough
- ~ Tour to Architecture diagram
- ~ Cost involved
- ~ End Notes
- ~ Structure overview
- ~ Data Ingestion
- ~ Data Validation
- ~ Data Transformation
- ~ Model Training and Tuning
- ~ Model Evaluation
- ~ Model Pusher
- ~ Training Pipeline
- ~ Frontend app design
- ~ Tour to the cloud and Service Overview

- ~ IAM setup
- ~ GCP setup
- ~ Workflow
- ~ Adding Self hosted runner
- ~ Conclude the project
- ~ Points to improve from current project
- ~ Assignments & External Resources

Raspberry Pi

Topic Name : K12

Sub-topic Name : CLASS10

Course link : <https://ineuron.ai/course/Raspberry-Pi>

Course Description :-

In this course, you will learn the basics of Raspberry pi and difference between microprocessor and microcontroller and how to use raspberry pi from which you can build amazing IOT applications with Raspberry PI OS. You will also learn Python programming language. This course will introduce the basic of Python library GPIO which will help you to start your journey in the field of artificial intelligence.

Course Features :-

- => Live instructor led classess
- => Completion certificate
- => Integrate academic knowledge with the tech
- => Real-time Project
- => Live Class Recording
- => Doubt Clearing
- => Assignment in all the Module
- => Quiz in every Module
- => Career Counselling
- => Completion Certificate

What you will learn :-

- => Get started with Raspberry Pi
- => Understanding Raspberry Pi
- => Understanding of Protocol used in Raspberry Pi
- => Basics of Electronic
- => OS Tour + Linux Fundamentals
- => Understanding Sensors and intergration with Raspberry Pi
- => Raspberry Programming
- => Raspberry Pi beginner projects

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Instructors :-

=> Sunny Bhaveen Chandra :

~ Sr. Data Scientist and lecturer at iNeuron.ai with working experience in computer vision, natural language processing and embedded systems. Hands-on experience leveraging machine learning, deep learning, transfer learning models to solve challenging business problems. Also, he has a vast interest in Robotics.

Curriculum details :-

- => Get started with Raspberry Pi :
 - ~ Introduction to microcontroller and microprocessor
 - ~ Microcontroller vs Microprocessor
 - ~ Example of microcontroller and microprocessor
 - ~ introduction to raspberry pi
 - ~ Various models of Raspberry Pi
 - ~ Comparison among Raspberry pi, Arduino, Nvidia Jetson Nano, Google coral
 - ~ History of Raspberry Pi
 - ~ Real life use cases for Raspberry Pi
 - ~ Daily Computation
 - ~ Internet of things
 - ~ AI development
 - ~ Purchase Raspberry Pi
 - ~ Ineuron innovation lab (One Nueron)
 - ~ Installtion of Raspberry Pi OS
 - ~ Configure and initiate initial boot of Operating System
 - ~ Get started with programming (C++ and Python)

=> Understanding Raspberry Pi :

- ~ *Raspberry Pi Architecture*
- ~ *Raspberry Pi specification*
- ~ *Raspberry Pi (40 Pin)*
- ~ *Components of Raspberry Pi*

=> Understanding of Protocol used in Raspberry Pi :

- ~ *Introduction to Protocol*
- ~ *UART, SPI, I2C, I2S, Digital I/O, wifi, and bluetooth*

=> Basics of Electronic :

- ~ *Boards*

=> Basics of Electronic :

- ~ *Basic Components (Resistor, Led, Transistor, Capacitor, Diode)*
- ~ *Basic Concepts electricity (Current, Power, voltage etc)*

=> OS Tour + Linux Fundamentals :

- ~ *Desktop Personalization*
- ~ *Working with Terminal*
- ~ *Raspberry Pi Terminal commands*
- ~ *Connecting to a Network*
- ~ *Remote Desktop*

=> Understanding Sensors and integration with Raspberry Pi :

- ~ *Introduction to Sensor*
- ~ *Difference between analog and digital sensors*
- ~ *Sensor list with use case*

=> Raspberry Programming :

- ~ *Supporting Languages*
- ~ *I/O Programming*
- ~ *GPIO configuration*
- ~ *GPIO programming*
- ~ *Interfacing of raspberry pi with various sensors*
- ~ *Interfacing analog and digital sensors with Raspberry Pi*

=> Raspberry Pi beginner projects :

- ~ *Camera Interfacing in Raspberry Pi*
- ~ *Configuration of camera module in Raspberry Pi*
- ~ *Integration of multiple camera*
- ~ *Installation of OpenCV*
- ~ *Real-time video streaming using Camera*

DBT

Topic Name : BIG DATA

Sub-topic Name : TECH STACK

Course link : <https://ineuron.ai/course/DBT>

Course Description :-

DBT data build tool helps data teams work like software engineers, transform data and control the flow to ship trusted data, faster.

DBT data build tool is an exciting tool in modern data manipulation, due to the shift from ETL to ELT in companies that rely on MPP databases in the cloud for example Snowflake, Redshift, Big query and others. this course will teach you the fundamentals of DBT data build tool. you will learn the structure of DBT data build tool and the main components.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => Connect DBT to Snowflake or another database
- => Create SQL transformations that use consistent logic
- => Learn DBT Best Practices

Requirements :-

- => System with minimum i3 processor or better
- => At least 4 GB of RAM
- => Working internet connection
- => Dedication to learn

Instructors :-

- => MD Imran :
 - ~ Working as Data Scientist with experience in solving real world business problems across different domains.

Curriculum details :-

- => DBT :
 - ~ What is DBT Preview
 - ~ DBT Cloud Account Creation
 - ~ Intro to Data Build Tool- create your first project
 - ~ DBT New project part 1 Preview
 - ~ DBT New project part 2
 - ~ DBT New Project part 3
 - ~ Snowflake connection
 - ~ Git push
 - ~ Adding raw sources to dbt project part 1
 - ~ Adding raw sources to dbt project part 2
 - ~ How dbt compile queries
 - ~ How to Write custom schemas
 - ~ How to test and debug dbt models
 - ~ Change the materialization
 - ~ Package management and dbt hub

Stats for Beginners

Topic Name : DATA SCIENCE

Sub-topic Name : STATS

Course link : <https://ineuron.ai/course/Stats-for-Beginners>

Course Description :-

If the goal of your career as a Data Scientist or Business Analyst then brushing up on your statistics skills is something you need to work on. But it's a difficult task to learn/re-learn all the stats seems like a daunting task. That's because we created this course. Here you will quickly get the absolutely essential stats knowledge for a Data Scientist or Analyst.

Course Features :-

- => Lifetime Dashboard
- => Free Course
- => Certificate
- => Assignment
- => Quiz

What you will learn :-

- => Understand what a Normal Distribution is
- => Explain the difference between continuous and discrete variables
- => Understand the Central Limit Theorem
- => Use the Z-Score and Z-Tables
- => Understand the difference between a normal distribution and a t-distribution
- => Create confidence intervals
- => Understand standard deviations
- => Understand what a sampling distribution is
- => Apply Hypothesis Testing for Proportions
- => Use the t-Score and t-Tables

Requirements :-

- => Basics math understanding

Instructors :-

=> krish naik :

~ Having 10+ years of experience in Data Science and Analytics with product architecture design and delivery. Worked in various product and service based Company. Having an experience of 5+ years in educating people and helping them to make a career transition.

Curriculum details :-

- => How to Learn Statistics for Data Science As A Self Starter- Follow My Way :
 - ~ Statistics Introduction Preview
- => Population vs Sample in Statistics
- => Gaussian distribution or Normal Distribution in statistics
- => Log Normal Distribution in Statistics
- => Covariance in Statistics
- => STATISTICS- Mean, Median And Mode Explained Easily
- => STATISTICS- Population VS Sample and its Importance
- => STATISTICS- What are Random Variables and its Types and its Importance?
- => STATISTICS- Gaussian/ Normal Distribution
- => STATISTICS- What is Central Limit Theorem?
- => STATISTICS- Chebyshev's Inequality
- => Statistics- What is Pearson Correlation Coefficient? Difference between Correlation and Covariance
- => Spearman's rank correlation coefficient- Statistics
- => Statistics-Finding Outliers in Dataset using Z- score and IQR
- => Standardization Vs Normalization- Feature Scaling

Django

Topic Name : WEB DEVELOPEMENT

Sub-topic Name : DJANGO

Course link : <https://ineuron.ai/course/Django>

Course Description :-

Django is a Python-based web framework. You'll be developing and learning about Django one step at a time in this course. We'll go through all you need to know about using Python, Django and other web technologies to create a website.

Course Features :-

- => Quizzes
- => Assignments
- => Hands-on practicals
- => Downloadable resources
- => Completion certificate

What you will learn :-

- => Django overview
- => Project implementation
- => Project deployment

Requirements :-

- => Basic knowledge of Python programming
- => A system with stable internet connection
- => Your dedication

Instructors :-

=> Khushali Shah :

~ A data scientist having rich experience working with MNCs and start-ups in the field of data science and machine learning. She has expertise in Chatbot development for various domains & been developing professionally for 6+ years with diverse job history. She also had positions in software module development, web app development, functional designs, requirement gathering, client interaction, and server setup/admin & can help everywhere in the stack; she loves wearing multiple hats to an extent. She also believes in enhancing her skills by training and learning new things day by day.

Curriculum details :-

=> Introduction :

~ Overview Preview

=> Django core :

- ~ Django features Preview
- ~ Creating a project
- ~ The development server
- ~ Creating the Polls app
- ~ Write your first view
- ~ `path()` argument: route
- ~ `path()` argument: view
- ~ `path()` argument: name

Class 7th Math

Topic Name : K12

Sub-topic Name : CLASS7

Course link : <https://ineuron.ai/course/Class-7th-Math>

Course Description :-

This course is useful for Grade 7 students. In this course, entire NCERT will be covered. Various questions from NCERT, NCERT exemplar and previous year will also be discussed. Dedicated doubt clearing sessions will also be conducted for helping the students regularly throughout their learning journey.

Course Features :-

- => Self Paced Videos
- => Completion Certificate

What you will learn :-

- => Algebra
- => Geometry
- => Statistics
- => Numbers
- => Mensuration

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Instructors :-

- => Jayant Topnani :
 - ~ Having 2+ years teaching experience and have mentored students online & offline across all boards.

Curriculum details :-

=> Integers :

- ~ Lecture 1 : Introduction Preview
- ~ Lecture 3 : NCERT Solutions Ex1.1 Question 2,3,4,5
- ~ Lecture 4 : Part 1 NCERT Solutions Ex1.1 Question 5,6,7,8,9,10
- ~ Lecture 5 : Integer Properties
- ~ Lecture 6 : NCERT Solutions Ex 1.2
- ~ Lecture 7 : Multiplication Property
- ~ Lecture 8 : NCERT Solutions Ex1.3 Question 1,2,3
- ~ Lecture 9 : NCERT Solutions Ex1.3 Question 3,4,5,6
- ~ Lecture 10 : NCERT Solutions Ex1.3 Question 7,8,9
- ~ Lecture 12 : NCERT Solutions Ex1.4
- ~ Lecture 13 : NCERT Solutions Ex1.4 Question 5,6,7

=> Fractions & Decimals :

- ~ Lecture 1 : Introduction NCERT Solutions Ex2.1 Question 1 & 2 Preview
- ~ Lecture 2 : NCERT Solutions Ex2.1 Question 3,4,5,6,7,8 Preview
- ~ Lecture 3 : Introduction Multiplication of Fractions
- ~ Lecture 4 : NCERT Solutions Ex2.2
- ~ Lecture 5 : Introduction NCERT Solutions Ex2.3
- ~ Lecture 6 : NCERT Solutions Ex2.3
- ~ Lecture 7 : Introduction NCERT Solutions Ex2.5
- ~ Lecture 8 : NCERT Solutions Ex2.4
- ~ Lecture 10 : NCERT Solutions Ex2.5 Question 1,2
- ~ Lecture 11 : NCERT Solutions Ex2.5 Question 4,5,6,7,8,9
- ~ Lecture 13 : NCERT Solutions Ex2.6
- ~ Lecture 15 : NCERT Solutions Ex2.7

=> Data Handling :

- ~ Lecture 1 : Data Handling Introduction
- ~ Lecture 2 : NCERT Solutions Ex3.1 Question 1,2,3,4,5,6,7,8,9
- ~ Lecture 3 : NCERT Solutions Ex3.2
- ~ Lecture 4 : NCERT Solutions Ex3.3 Question 1,2
- ~ Lecture 6 : NCERT Solutions Ex3.3 Question 4
- ~ Lecture 9 : NCERT Solutions Ex3.4

=> Simple Equations :

- ~ Lecture 1 : Introduction
- ~ Lecture 2 : NCERT Solutions Ex4.1
- ~ Lecture 3 : Introduction Solving Equations
- ~ Lecture 4 : NCERT Solutions Ex 4.2

- ~ Lecture 5 : NCERT Solutions Ex4.3
- ~ Lecture 6 : Introduction NCERT Solutions Ex4.4
- ~ Lecture 7 : NCERT Solutions Ex 4.4

=> Lines and Angles :

- ~ Lecture 1 : Some Important Definitions : Line segment, Line, Angles, Acute Angle, Obtuse Angle, Reflex Angle, Straight Angle, Complementary Angle, Supplementary Angle
- ~ Lecture 2 : Adjacent Angle, Linear Pair, Vertically Opposite Angle
- ~ Lecture 3 : Ex 5.1 Q 1 to 9
- ~ Lecture 4 : Ex 5.1 Q 9 to 14
- ~ Lecture 5 : Intersecting Lines, Transversal and Angles made by a Transversal, Transversal of Parallel Lines.
- ~ Lecture 6 : Checking for Parallel Lines, Ex 5.2

=> The Triangle and its Properties :

- ~ Lecture 1 : Introduction
- ~ Lecture 2 : Elements of Triangle, Vertex, Sides and Angles
- ~ Lecture 3 : Classification of Triangles on the basis of Sides and on the basis of Angles, Angle Sum Property of a Triangle
- ~ Lecture 4 : Medians and Altitudes of a Triangle, Ex 6.1
- ~ Lecture 5 : Exterior Angles of a Triangle and its Properties, Ex 6.2
- ~ Lecture 6 : Ex 6.3
- ~ Lecture 7 : Two Special Triangles, Equilateral and Isosceles
- ~ Lecture 8 : Sum of Lengths of Two Sides of a Triangle
- ~ Lecture 9 : Ex 6.4, Q 1 to 5
- ~ Lecture 10 : Right-Angled Triangles and Pythagoras Property
- ~ Lecture 11 : EXERCISE 6.5 Q 3 to 8

=> Congruence of Triangles :

- ~ Lecture 1 : Congruent Figures, Congruence of Triangles
- ~ Lecture 2 : Ex 7.1 Q 1 to 4
- ~ Lecture 3 : Criteria for Congruence of Triangles, SSS Criteria
- ~ Lecture 4 : SAS, ASA Congruence Condition
- ~ Lecture 5 : Congruence Among Right Angle Triangle (RHS Congruence Condition)
- ~ Lecture 6 : Ex 7.2 Q 1 to 5
- ~ Lecture 7 : Ex 7.2 Q 5 to 6

=> Comparing Quantities :

- ~ Lecture 1 : Introduction, Understanding Ratio, Equivalent Ratio
- ~ Lecture 2 : Unitary Method, Ex 8.1, Proportion
- ~ Lecture 3 : Percentage, another way of Comparing Quantities
- ~ Lecture 4 : Converting Decimal to Percentage
- ~ Lecture 5 : NCERT Ex 8.2 Q 6 to 8
- ~ Lecture 6 : Profit and Loss
- ~ Lecture 7 : Question Practice on Profit and Loss
- ~ Lecture 8 : Charge Given on Borrowed money / Simple Interest
- ~ Lecture 9 : EX 8.3 Q 1 to 7
- ~ Lecture 10 : EX 8.3 Q 8 to 11

=> Rational Numbers :

- ~ Lecture 1 : Introduction, Positive and Negative Rational number, Three Important Properties of Rational Numbers, Equivalent Number
- ~ Lecture 2 : Representing Rational Number on Number Lines
- ~ Lecture 3 : Rational Number in Standard Form
- ~ Lecture 4 : Comparison of Rational Numbers, Rational Numbers between two Rational Numbers
- ~ Lecture 5 : EX 9.1 Q 1 to 5
- ~ Lecture 6 : EX 9.1 Q 6 to 10

=> Practical Geometry :

- ~ Lecture 1 : Construct line parallel to given line & triangle if 3 sides are given
- ~ Lecture 2 : Construction triangle SAS, ASA & RHS

=> Perimeter and Area :

- ~ Lecture1_Introduction_&_Course_Content
- ~ Lecture2_Square_&_Rectangle
- ~ Lecture3_NCERT_EX_11.1_PROBLEM_DISCUSSION
- ~ Lecture4_Triangle_As_Part_Of_Rectangle
- ~ Lecture5_Area_Of_Parallelogram
- ~ Lecture6_Area_Of_Triangle
- ~ Lecture7_NCERT_EX11.2_PROBLEM_DISCUSSION
- ~ Lecture8_Circle_Circumference_&_Area_Part1
- ~ Lecture8_Circumference_&_Area_Circle_Part2
- ~ Lecture8_Circumference_&_Area_Circle_Part3
- ~ Lecture9_NCERT_EX11.3_PROBLEM_DISCUSSION
- ~ Lecture10_CONVERSION_OF_UNITS
- ~ Lecture11_Applications_Perimeter_&_Area
- ~ Lecture12_NCERT_EX11.3_PROBLEM_DISCUSSIONS

=> Algebraic Expressions :

- ~ Lecture1_Introduction_&_Topics
- ~ Lecture2_Algebraic_Terminologies
- ~ Lecture3_Like_Vs_Unlike_Terms
- ~ Lecture4_NCERT_EX_12.1_PROBLEM_DISCUSSION
- ~ Lecture5_Addition_&_Subtraction_Algebraic_Expressions
- ~ Lecture6_NCERT_EX_12.2_PROBLEM_DISCUSSION
- ~ Lecture7_Finding_Value_Of_Algebraic_Expression
- ~ Lecture8_NCERT_EX_12.3_PROBLEM_DISCUSSION
- ~ Lecture9_Number_Pattern_Rules
- ~ Lecture10_NCERT_EX12.4_PROBLEM_DISCUSSIONS
- ~ Lecture11_Chapter_Summary_The_End

=> Exponents and Powers :

- ~ Lecture1_Course_Content_&_Introduction

- ~ Lecture2_Exponents_&_Powers
- ~ Lecture3_NCERT_EX_13.1_PROBLEM_DISCUSSIONS
- ~ LECTURE4_LAWS_OF_EXPONENTS
- ~ LECTURE5_NCERT_13.2_PROBLEM_DISCUSSION
- ~ LECTURE6_STANDARD_FORM_NOTATION
- ~ LECTURE7_NCERT_EX13.3_PROBLEM_DISCUSSION

=> Symmetry :

- ~ Lecture1_Introduction_To_Symmetry
- ~ Lecture2_Line_Of_Symmetry_For_Regular_Polygon
- ~ Lecture3_NCERT_EX_14.1_PROBLEM_DISCUSSION
- ~ Lecture4_All_about_Rotational_Symmetry
- ~ Lecture5_NCERT_EX_14.2_PROBLEM_DISCUSSION
- ~ Lecture6_Line_&_Rotational_Symmetry
- ~ Lecture7_NCERT_EX_14.3_PROBLEM_DISCUSSION

=> Visualising Solid Shapes :

- ~ Lecture1_Introduction_&_Course_Walkthrough
- ~ Lecture2_All_About_Nets
- ~ Lecture3_NCERT_EX15.1_PROBLEMS_DISCUSSION
- ~ Lecture4_Oblique_Vs_Isometric_Sketches
- ~ Lecture5_NCERT_EX15.2_PROBLEM_DISCUSSION
- ~ Lecture6_Visualizing_Solid_Objects
- ~ Lecture7_Viewing_Different_Sections_Of_Solid

Class 10 Biology

Topic Name : K12

Sub-topic Name : CLASS10

Course link : <https://ineuron.ai/course/Class-10-Biology>

Course Description :-

Biology is the study of life. It is the study of living organisms and how they interact with the environment. Biology recognizes the cell as the basic unit of life, genes as the unit of heredity and evolution as an engine which boosts the formation of new species. The study of life has helped in shaping the world. It has credible answers to why things happen in a scientific manner. iNeuron allows you to explore all the content from NCERT physics in a proper manner and flow.

Course Features :-

- => Self paced video session
- => Covered entire class 10th Biology syllabus
- => Solved questions chapter wise
- => Notes
- => Previous year solved questions

What you will learn :-

- => Entire NCERT Class 10th Biology Syllabus
- => Chapter wise solution with detailed explanation

Requirements :-

- => Computer with Internet Connectivity

Curriculum details :-

=> LIFE PROCESSES :

- ~ Nutrition & its types
- ~ Respiration
- ~ Transport
- ~ Excretion

=> CONTROL & COORDINATION :

- ~ I. Tropic movements in plants
- ~ II. Introduction of plant hormones
- ~ III. Control and co-ordination in animals
- ~ IV. Nervous system
- ~ V. Voluntary, involuntary and reflex action
- ~ VI. Chemical co-ordination
- ~ VII. Animal Hormones

=> HOW DO ORGANISMS REPRODUCE :

- ~ I. Reproduction in animals and plants (asexual and sexual)
- ~ II. Types / Modes of Reproduction
- ~ III. Safe sex vs HIV/AIDS
- ~ IV. Child bearing and womens health

=> HEREDITY & EVOLUTION :

- ~ I. Heredity
- ~ II. Mendels contribution- Laws for inheritance of traits
- ~ III. Basic concepts of evolution
- ~ IV. Evolution
- ~ V. Sex determination
- ~ VI. Speciation

=> OUR ENVIRONMENT :

- ~ I. Eco-system
- ~ II. Environmental problems
- ~ III. Ozone depletion
- ~ IV. Waste production and their solutions
- ~ V. Biodegradable and non-biodegradable substances

=> SUSTAINABLE MANAGEMENT OF NATURAL RESOURCES :

- ~ I. Conservation and judicious use of natural resources
- ~ II. Forest and wild life
- ~ III. Coal and Petroleum conservation
- ~ IV. Big dams - advantages and limitations; alternatives
- ~ V. Harvesting of water
- ~ VI. Sustainability of natural resources

Angular JS

Topic Name : WEB DEVELOPEMENT

Sub-topic Name : ANGULAR JS

Course link : <https://ineuron.ai/course/Angular-JS>

Course Description :-

Angular is a platform and framework for building single-page client applications using HTML and TypeScript. Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your apps. AngularJS is a client side JavaScript MVC framework to develop a dynamic web application. AngularJS was originally started as a project in Google but now, it is open source framework.

Course Features :-

- => Lifetime Dashboard
- => Free Course
- => Certificate
- => Assignment
- => Quiz

What you will learn :-

- => Practical implementation of Angular JS in real world
- => End to End concepts understanding

Requirements :-

- => Computer with Internet connectivity
- => Basic Programming understanding

Instructors :-

- => Keshav Singh :
- ~

Curriculum details :-

- => Introduction About Angular JS (Hindi) :
 - ~ *Introduction Preview*
- => Course Structure Breakup and Environment Setup 2 Angular JS (Hindi)
- => Components in Angular JS (Hindi)
- => Bindings In Angular (Hindi)
- => Bindings in Angular JS Part-2 (Hindi)
- => Communication Between Components - Angular JS | Hindi
- => Two Way Data Binding Angular JS | Hindi

Data Analysis Using Python

Topic Name : K12

Sub-topic Name : CLASS10

Course link : <https://ineuron.ai/course/Data-Analysis-Using-Python>

Course Description :-

In this course, you will learn to extract insights from the data. This course is designed to help you deal with data analysis and data manipulation using the popular python library Pandas. You will learn powerful functions to present the facts from data in the most straightforward and accurate way.

Course Features :-

- => Online Instructor-led learning
- => Practical Implementation
- => Integrate academic knowledge with the tech
- => Real-time Project
- => Live Class Recording
- => One to One Doubt Clearing
- => Assignment in all the Module
- => Quiz in every Module
- => Career Counselling
- => Completion Certificate

What you will learn :-

- => Introduction to Data Analysis
- => Importance of Data
- => Integrated development environment
- => Python Packages
- => Python Libraries
- => File Formats
- => Pandas Library
- => Data Cleaning
- => Data Manipulation
- => Pandas Functions
- => Feature engineering

Requirements :-

- => Interest to learn
- => Dedication
- => System with good internet connection

Curriculum details :-

- => Introduction to the Course :
 - ~ Course Introduction
 - ~ Who is this course for?
 - ~ Course overview & course outcome
 - ~ What is Data? How data will be collected?
 - ~ Why Data is important?
 - ~ What is DataFrame?
 - ~ Why DataFrame is used?
 - ~ What is Analysis?
 - ~ Why Analysis is used?
 - ~ What is Data Analysis?
 - ~ Why Data Analysis is used?
- => Assignment 1 :
 - ~ Is it possible to create and innovate things without data?
- => System setup :
 - ~ What is IDE?
 - ~ Why IDE is used?
 - ~ Advantages of using an IDE?
 - ~ Google Colab

=> PyPI :

- ~ What is PyPI?
- ~ What is pip?
- ~ Installing your first library using pip

=> Packages :

- ~ What is a Package?
- ~ How to install Python Packages

=> Library :

- ~ What is a library?
- ~ What is the difference between package and library?

=> Open Source :

- ~ What do you mean by open-source community?
- ~ What is an open-source package or a library?
- ~ Why is it important to do open-source contribution

=> File Formats :

- ~ What is a file?
- ~ What are the types of files?
- ~ .csv
- ~ .txt
- ~ .json
- ~ .xlsx

=> Pandas :

- ~ What is pandas library?
- ~ Alternatives of pandas
- ~ Why pandas use ?

=> Cleaning Data Using Pandas :

- ~ What is Data cleaning?
- ~ Why it is necessary to perform Data cleaning?
- ~ Performing Data cleaning using pandas
- ~ Removing unnecessary columns in the DataFrame
- ~ Skipping unnecessary rows in a CSV file
- ~ Changing the index of a DataFrame.
- ~ Renaming columns to a more recognizable set of labels.

=> Pandas Functions :

- ~ What do you mean by functions?
- ~ What are Pandas functions?
- ~ Important Pandas functions
- ~ Practical: read_csv()
- ~ Practical: head()
- ~ Practical: describe()
- ~ Practical: memory_usage()
- ~ Practical: astype()
- ~ Practical: loc[:]
- ~ Practical: value_counts()
- ~ Practical: groupby()

=> Assignment 2 :

- ~ Download any dataset of your choice and try to apply these pandas functions on your own.
- ~ Try to find out more functions in pandas and try to implement them.

=> Feature Engineering :

- ~ What do you mean by feature?
- ~ What is Feature Engineering?
- ~ What do you mean by Feature Transformations?
- ~ Practical: Transforming columns into the same scale
- ~ What do you mean by Feature Extraction?
- ~ Practical: Extracting year from your birthdate
- ~ What do you mean by Feature selection?
- ~ Practical: selecting the important features from the dataset

=> Project :

- ~ Take any dataset and perform data analysis using pandas

=> Course Summary :

- ~ Course outro
- ~ Future learning path
- ~ Order by clause

House Price Prediction

Topic Name : DATA SCIENCE

Sub-topic Name : MACHINE LEARNING PROJECT

Course link : <https://ineuron.ai/course/House-Price-Prediction>

Course Description :-

The selling price of a property in a specific area can be determined with the use of house price predictions, and consumers can choose the ideal moment to purchase a home. In this project, "House Price Prediction Using Machine Learning," our goal is to develop a machine learning model to forecast house prices in the State of California using data from the census.

Course Features :-

- => Do Everything In Industry Grade Lab
- => Learn As Per Your Timeline
- => Hands-On Industry Real-Time Projects.
- => Self Paced Learning
- => Dashboard Access

What you will learn :-

- => Real Time Projects
- => House Price Prediction
- => Preparing Dataset And Basic Analysis
- => Preparing Dataset For Model Training
- => Training the Model
- => Performance Metrics
- => Creating A Flask Web Application
- => Deployment

Requirements :-

- => System with minimum i3 processor or better
- => At least 4 GB of RAM
- => Working internet connection
- => Dedication to learn

Instructors :-

=> krish naik :

~ Having 10+ years of experience in Data Science and Analytics with product architecture design and delivery. Worked in various product and service based Company. Having an experience of 5+ years in educating people and helping them to make a career transition.

Curriculum details :-

- => Welcome to the Course :
 - ~ Course Overview
 - ~ Dashboard Introduction
- => Project :- House Price Prediction :
 - ~ Understanding the dataset
 - ~ Preparing Dataset And Basic Analysis
 - ~ Preparing Dataset For Model Training
 - ~ Training the Model
 - ~ Performance Metrics
 - ~ Prediction Of New Data
 - ~ Pickling the model File
 - ~ Setting up Github And VS Code
 - ~ Tools And Softwares Required
 - ~ Creating A New Environment
 - ~ Setting up Git
 - ~ Creating A Flask Web Application
 - ~ Running And Testing Our Application
 - ~ Prediction From Front End Application
 - ~ Procfile For Heroku Deployment
 - ~ Deploying App To Heroku
 - ~ Deploying the App Using Dockers

Explainable AI

Topic Name : DATA SCIENCE

Sub-topic Name : MACHINE LEARNING

Course link : <https://ineuron.ai/course/Explainable-AI>

Course Description :-

Explainable AI

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => All about XAI
- => Explaining AI with Python
- => West Nile virus a case of life or death
- => Explaining Machine Learning with Facets

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Instructors :-

=> Rishav Dash :

~ This is Rishav Dash. I am a Jr. Data Scientist and mentor at INeuron.ai with working experience in computer vision, natural language processing, Machine Learning, and AOps. Hands-on experience leveraging machine learning, deep learning, transfer learning models to challenging real-world problems, and building products to solve peoples problems.

Curriculum details :-

=> Explainable AI :

- ~ Introduction to Explainable AI (XAI)
- ~ All about XAI
- ~ Explaining AI with Python
- ~ West Nile virus a case of life or death
- ~ XAI can save lives using Google Location H
- ~ Explaining Machine Learning with Facets
- ~ Microsoft Azure ML Model Interp SHAP
- ~ SHAP Implementation
- ~ Building XAI solution from scratch
- ~ AI fairness with Google_s What-if-Tool(WIT)
- ~ Local Interpretable Model-Agnostic Explanation(LIMEI)
- ~ The END

Streamlit

Topic Name : DATA SCIENCE

Sub-topic Name : MACHINE LEARNING

Course link : <https://ineuron.ai/course/Streamlit>

Course Description :-

In this course, you will learn about the fundamental concepts related to the Streamlit library. After completion of this course, you will be able to create Web applications for your Data science models with the help of Streamlit Library.

Course Features :-

- => Practical Implementation
- => Downloadable resources
- => Class Recordings
- => Quiz Questions
- => Completion Certificate

What you will learn :-

- => Streamlit
- => Web-apps
- => Building UI

Requirements :-

- => Prior Knowledge of Python Programming Language
- => Interest to learn
- => Your dedication

Instructors :-

=> Jaydeep Dixit :

~ Jaydeep Dixit is a data scientist and Blockchain Developer working at iNeuron having 1.5+ years of total experience. He specializes in Machine Learning and Blockchain. He has worked on various end-to-end projects in both machine learning and Blockchain. In addition to his primary job function, he has been recognized for his problem-solving skills.

Curriculum details :-

=> Course Introduction :

- ~ Introduction to Streamlit Preview
- ~ Who is this course for?
- ~ Course overview Preview

=> Installation :

- ~ Installation of Streamlit

=> Main Concepts :

- ~ Basic concepts
- ~ apiReference Preview
- ~ Data display elements
- ~ Chart elements
- ~ Input widgets
- ~ Media elements
- ~ Layout container
- ~ Status elements
- ~ Control flow

=> Project :

- ~ Banknote Authentication

=> Course Summary :

- ~ Summary
- ~ Future learning

The Pro Backend Developer

Topic Name : WEB DEVELOPEMENT

Sub-topic Name : FULL STACK WEB DEVELOPMENT

Course link : <https://ineuron.ai/course/The-Pro-Backend-Developer>

Course Description :-

This course is titled pro for a reason. In this practical hands-on course, you will learn how to build complex backend applications that can be used for any web or mobile application. Your REST API will be in production with docs, social logins, images, authentications, mail and, much more. This is a true pro backend course.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => MongoDB
- => Heroku Cloud
- => Swagger
- => Authentication
- => File, image and form handling
- => MORGAN and razorpay
- => Configs and imports
- => Controllers and routes

Requirements :-

- => System with minimum i3 processor or better
- => At least 4 GB of RAM
- => Working internet connection
- => Dedication to learn

Instructors :-

=> Hitesh Choudhary :

~ I like to make videos related to code and tech in my free time. I also lead a few tech teams in startups, help in hiring talent for companies. I am also on a part time traveller, with 31 countries checked off so far!

Curriculum details :-

=> Getting started :

- ~ Goal of this course and instructions
- ~ Tools for backend developer
- ~ MongoDB MAC install
- ~ MongoDB WIN install
- ~ MongoDB in cloud - Atlas
- ~ Mongo GUI - compass

=> Take it up to Heroku - Production :

- ~ Things you need to deploy on Heroku
- ~ Plan your application
- ~ Types of web request
- ~ Framework - Express, Koa, Hapi
- ~ Starting with package JSON file
- ~ Your first express app
- ~ Request Response and Status code
- ~ All social routes
- ~ Handle the date situation
- ~ Parameters and bugs in route
- ~ Pushing app to HEROKU
- ~ Debug social app in production

=> Swagger Docs :

- ~ What is swagger and api docs
- ~ Nodemon ext and YAML docs

- ~ Authentication token for swagger docs
- ~ Docs for HTTP methods swagger
- ~ A new documentation centric project
- ~ Setup information - swagger
- ~ Authentication and Authorization - swagger
- ~ String based GET request - swagger
- ~ handling objects - swagger
- ~ handling array in Swagger docs
- ~ Sending data in URL - swagger
- ~ managing request body in swagger
- ~ handle url query in swagger
- ~ handling images in swagger
- ~ handling header tokens in swagger

=> Authentication :

- ~ What we have done till section 3 - backend
- ~ Hiding secrets in backend
- ~ Picking up a database for backend
- ~ Why we need mongoose - ODM
- ~ Pro db modeling tools
- ~ Creating model for auth system
- ~ Creating basic structure for auth system
- ~ Creating user schema and dotenv
- ~ Registering a user in auth system
- ~ Database connection in auth system
- ~ What is a middleware
- ~ Handling password situation
- ~ What is JWT and creating token
- ~ Register route in auth app
- ~ Login flow for auth app
- ~ Web vs Mobile
- ~ Writing custom middleware
- ~ Setting up secure cookies

=> File, image and form handling :

- ~ Why people face issue in image upload
- ~ Cloudinary and EJS
- ~ How GET works and postman issues
- ~ Using template engines
- ~ Biggest confusion in front end forms
- ~ Handling images in forms
- ~ Handling images in forms part 2
- ~ upload image to cloudinary or other providers
- ~ Handling multiple files and uploading them

=> Theory and Razorpay :

- ~ File structure for production app
- ~ Getting a logger - MORGAN
- ~ Error handler and Promises
- ~ Sending emails using nodemailer
- ~ Why mongoose docs are important
- ~ Razorpay project
- ~ Razorpay front end integration

=> Big Ecommerce app starts :

- ~ Project requirement
- ~ User modeling and file structure
- ~ Product model discussion
- ~ Order Model discussion
- ~ How forgot password feature work
- ~ Functions in user model and hooks

=> Basic Config and imports :

- ~ Getting files and folders ready
- ~ Preparing basic express app
- ~ Routes and controllers in dummy
- ~ Injecting docs and middleware
- ~ Custom error handlers
- ~ The big Promise

=> User model and signup :

- ~ Creating a user model and validator
- ~ password encryption and mongoose prototypes
- ~ Validating the password
- ~ creating JWT tokens
- ~ forgot password and crypto hashing
- ~ User routes and postman
- ~ Signup a user and cookies
- ~ Database connection
- ~ Testing the user signup with postman
- ~ Handling image upload
- ~ Testing photo upload and user signup
- ~ yes, we know about postman files

=> User controllers and routes :

- ~ Login route and controller
- ~ logout controller and route
- ~ Send email from node
- ~ Forgot password controller
- ~ Reset password controller and routes
- ~ Middleware - injecting information

- ~ User dashboard controller and routes
- ~ Update the password for a user
- ~ Updating the user profile
- ~ User, admin, manager and more roles
- ~ Manager only routes
- ~ Admin get a single user
- ~ Admin can update any user
- ~ Admin can delete a user now

=> Working on Product Model :

- ~ Product middleware setup for routes
- ~ Product Model and refs
- ~ A long talk on URL replace and mongo operators
- ~ Creating a product
- ~ Where clause in search
- ~ Where clause Pager
- ~ Aggregation filter in Where Clause
- ~ Get all products with WHERE and pager
- ~ Debugging and testing of product add and get

=> More routes in Products :

- ~ Single product route
- ~ Update the product with photos
- ~ Delete a product and minor bug
- ~ Testing and debugging
- ~ Add a review
- ~ Delete a review and requested routes
- ~ Configure routes for reviews

=> Razorpay and Stripe :

- ~ Stripe Docs
- ~ Stripe controllers
- ~ Razorpay payments and order
- ~ Setup payment routes

=> Processing Orders :

- ~ Order model in action
- ~ Creating a order and BSON
- ~ Testing create order and routes
- ~ Populate fields in order
- ~ Order of routes is important
- ~ Updating the stock
- ~ Delete order and push to git
- ~ Pushing code to production server

=> OAuth and Social Logins :

- ~ Social login foundation and demo app
- ~ Consent screen and API keys
- ~ Why passport.js
- ~ Package installation
- ~ Home routes and EJS
- ~ Preparing routes for login
- ~ Showing consent screen of google
- ~ Getting information and email from google
- ~ Moving google data to database
- ~ Serialize and deserialize user
- ~ Protect the Home

Pro Aptitude - C++

Topic Name : APTITUDE

Sub-topic Name : APTITUDE

Course link : <https://ineuron.ai/course/Pro-Aptitude---C++>

Course Description :-

This course is designed mostly for C++ test takers.

Course Features :-

=> Quizzes

=> Course completion certificate

What you will learn :-

=> C++ Theoretical Test

=> C++ Practical Test

=> C++ Aptitude Test

Requirements :-

=> System with minimum i3 processor or better

=> At least 4 GB of RAM

=> Working internet connection

=> Dedication to solve

Curriculum details :-

=> C++ Coding Test :

~ C++ Test 1

~ C++ Test 2

~ C++ Test 3

~ C++ Test 4

Getting started with Cloud

Topic Name : K12

Sub-topic Name : CLASS10

Course link : <https://ineuron.ai/course/Getting-started-with-Cloud>

Course Description :-

This course will assist students in grasping the basic concepts of cloud computing with its various applications. This course will equip students with the proper start that they require to begin their career in cloud computing in today's modern environment, where cloud engineers are in high demand. This hands-on practical-oriented course will enable students to apply their cloud computing skills and help them in starting with a lucrative career in cloud computing.

Course Features :-

- => Online Instructor-led learning
- => Practical Implementation
- => Integrate academic knowledge with the tech
- => Real-time Project
- => Live Class Recording
- => Doubt Clearing
- => Assignment in all the Module
- => Quiz in every Module
- => Career Counselling
- => Completion Certificate

What you will learn :-

- => Cloud computing fundamentals
- => Types of Clouds
- => Risks
- => Features of Cloud Computing
- => Virtualization
- => Characteristics of Cloud
- => Cloud Computing Architecture

Requirements :-

- => Interest to learn
- => Dedication
- => System with good internet connection

Curriculum details :-

- => Course Introduction :
 - ~ Course introduction
 - ~ Who is this course for?
 - ~ Course overview & course outcome
 - ~ Course Pre-requisite
 - ~ What is Cloud?
 - ~ What is Cloud computing?
 - ~ Why cloud is important?
 - ~ History of Cloud computing
- => Assignment 1 :
 - ~ What are the benefits of Cloud?
 - ~ Write down the benefits of cloud computing?
- => Basic Cloud concepts :
 - ~ Types of Cloud
 - ~ What is Public cloud?
 - ~ What are the different types of Public cloud?
 - ~ What is a Private cloud?
 - ~ What are the different types of Private cloud?
 - ~ What is Hybrid cloud?
 - ~ What are the different types of Hybrid cloud?
 - ~ What is Community cloud?
 - ~ What are the different types of Community cloud?
 - ~ Service Models
 - ~ What is Infrastructure as a Service (IaaS)
 - ~ What is Platform as a Service (PaaS)

~ What is Software as a Service (SaaS)

=> Assignment 2 :

~ Name the clouds which are most popular?

=> Assignment 3 :

~ What is the difference between IaaS, PaaS, and SaaS?

=> Risks :

~ What is Security?

~ Why is Security important?

~ What is Privacy?

~ Why is Privacy important?

~ What is LOCK-IN?

~ Why is LOCK-IN risky?

~ What is Isolation failure?

~ What is Management Interface compromise?

~ What is Insecure or Incomplete data deletion?

=> Assignment 4 :

~ How to avoid fraudulent activities on the cloud?

=> Virtualization :

~ What is Virtualization?

~ What is the intuition of Virtualization?

~ Types of Virtualization

~ What is Hardware virtualization?

~ What is the use of Hardware virtualization?

~ What is Operating system virtualization?

~ What is the use of the Operating system virtualization?

~ What is Server virtualization?

~ What is the use of Server virtualization?

~ What is Storage virtualization?

~ What is the use of Storage virtualization?

=> Assignment 5 :

~ How does virtualization work in Cloud computing?

=> Data Virtualization :

~ What is Data virtualization?

~ What are the advantages of Data virtualization?

~ What are the disadvantages of Data virtualization?

~ What are the uses of Data virtualization?

~ Data virtualization tools

=> Assignment 6 :

~ How well do you know about Data virtualization?

=> Characteristics :

~ Essential Characteristics

~ On-demand self-service

~ Broad network access

~ Resource pooling

~ Rapid elasticity

~ Measured service

~ Common Characteristics

~ Massive scale

~ Homogeneity

~ Virtualization

~ Low cost software

~ Resilient computing

~ Geographic distribution

~ Service orientation

~ Advanced security

=> Assignment 7 :

~ What is the difference between Essential and Common characteristics?

=> Cloud Computing-Architecture :

~ What is Front End?

~ What is Back End?

~ What is Client infrastructure?

~ What are the Applications?

~ What are the Services?

~ Software as a Service (SaaS)

~ Example of Software as a Service

~ Platform as a Service (PaaS)

~ Example of Platform as a Service

~ Infrastructure as a Service (IaaS)

~ Example of Infrastructure as a Service

~ What is the Runtime cloud?

~ What is Storage?

~ What is Infrastructure?

~ What is Management?

~ What is Security?

~ What is Internet?

=> Assignment 8 :

~ Describe the cloud computing architecture in your own words?

=> Course Summary :

~ Course outro

~ Future learning path

GIT

Topic Name : DEVOPS

Sub-topic Name : GIT

Course link : <https://ineuron.ai/course/GIT>

Course Description :-

The Git course will teach you how to utilise the Git version management system in a hands-on manner. Git is a collaborative file management system for large and small projects. As a result, the team can develop its product regularly.

Course Features :-

- => Source code
- => Downloadable resources
- => Quizzes
- => Completion certificate

What you will learn :-

- => Git Introduction
- => Git Commands
- => Git Branching
- => Merging
- => Tagging
- => Rebasing

Requirements :-

- => Prior knowledge of Linux
- => A System with good internet connection
- => Your dedication

Instructors :-

=> Sourangshu Pal :

~ Visual Computing Engineer and instructor at iNeuron.ai having 3 years of diverse experience in the discipline of visual computing with specialization in Deep Learning and Computer Graphics. Loves to analyze, process, and model visual data then interpret the insights to create actionable plans for solving challenging business problems.

Curriculum details :-

=> Git Introduction :

- ~ Git Introduction Preview
- ~ What is Version Control?
- ~ Types of Version Control
- ~ What is Git?
- ~ Why Git?
- ~ Git Installation in Windows
- ~ Git Installation in Linux
- ~ Git Setup
- ~ Git Terminologies

=> GIT Repository :

- ~ Repositories in GIT Preview
- ~ Creating Repository
- ~ Checking Repository History
- ~ Doing Commits
- ~ Git Diff
- ~ Git Restore
- ~ Git Ignore

=> Git Commands :

- ~ Tagging Preview
- ~ Branching
- ~ Branching Practicals
- ~ Merging
- ~ Merge Conflicts

=> Git Branching :

- ~ Remote Repository
- ~ Cloning Repository
- ~ Working with Remote Repository
- ~ Pushing to Remote Failed in Github
- ~ Personal Access Token Setup in Windows
- ~ Personal Access Token Setup in Linux
- ~ Pull Request
- ~ GIT Fetch & Pull

~ *Fork*

=> **Rebasing :**

~ *Rebasing*

~ *Interactive Rebasing*

~ *Git Rewrite History*

~ *Git Rewrite History continued*

~ *Cherry Picking*

=> **Outro :**

~ *Modify Recent Commits*

~ *Git Revert*

~ *Git Checkout*

~ *Git Reset*

~ *Git Stash*

~ *Git Reflog*

~ *Course Outro*

Full Stack Blockchain Development

Topic Name : BLOCKCHAIN

Sub-topic Name : BLOCKCHAIN MASTERS

Course link : <https://ineuron.ai/course/Full-Stack-Blockchain-Development>

Course Description :-

Full Stack Blockchain Development course is a live mentor-led certification program with by iNeuron. In this course you will learn the entire stack required to work in Permissionless Blockchain development. This course focuses on latest Blockchain industry standards like Ethereum Blockchain, Solidity, Decentralized Autonomous Organisations, Decentralized Finance, Non Fungible Tokens, Polygon Network, Polkadot Blockchain, Oracles along with complete development stack in Javascript and many more Blockchain concepts.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => Web 1.0 vs Web 2.0 vs Web 3.0
- => What is Blockchain technology?
- => Bitcoin Blockchain
- => Ethereum Blockchain
- => Solidity
- => Oracles
- => DAO
- => DeFi
- => NFT
- => Layer 2 Blockchain
- => Truffle Suite
- => Hardhat
- => Polkadot

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Instructors :-

=> krish naik :

~ Having 10+ years of experience in Data Science and Analytics with product architecture design and delivery. Worked in various product and service based Company. Having an experience of 5+ years in educating people and helping them to make a career transition.

=> Sanjeevan Thorat :

~ Data Scientist and Blockchain developer, with experience in developing and managing end to end solutions. I have hands-on experience in Python Programming Language, Machine Learning Deep Learning and Natural language processing. Blockchain development experience in smart contracts, Decentralised Finance applications, DAOs, NFTs and Oracles running on Ethereum and Polygon blockchains. I have worked with various clients for different industry projects in the blockchain space. I specialize in building smart contracts on the Ethereum blockchain along with JavaScript integration for enhancing user experience to generate maximum returns on investment.

=> Navin Reddy :

~ I am Corporate Java trainer. Since past few years successfully trained many professionals at JP Morgan, Accenture, Polaris and L&T infotech. My youtube channel "Telusko" presently has 1.7 million subscribers. Passionate about Java Technology for over a decade and moved on as a corporate trainer. I am certified blockchain developer and Currently, building Applications running on Blockchain (dapps).

Curriculum details :-

=> Course Introduction :

- ~ Course overview
- ~ A brief history of internet technologies
- ~ Web 1.0 vs Web 2.0 vs Web 3.0

- ~ *What is Blockchain technology?*
- ~ *Why do we need Blockchain technology?*
- ~ *The connected world and the Blockchain: A disruptive computing paradigm*
- ~ *Centralized vs Decentralized networks*
- ~ *Distributed Systems overview*

=> **Web Development :**

- ~ *What is Web Development?*
- ~ *Client-Server Architecture*
- ~ *What are APIs?*
- ~ *What is Front-end web development?*
- ~ *What is Back-end web development?*
- ~ *Components of Full-Stack Web Development Applications*

=> **HTML :**

- ~ *How do websites work?*
- ~ *Preview*
- ~ *HTML vs CSS vs Javascript*
- ~ *HTML files*
- ~ *Doctype & HTML Boilerplate*
- ~ *Spaces & Line Breaks*
- ~ *Heading Tag*
- ~ *Paragraph & Pre Tag*
- ~ *Difference between Elements, Attributes & Tags*
- ~ *Comments*
- ~ *Useful Tags*
- ~ *Nesting of Tags*
- ~ *Extensions in HTML*
- ~ *Live Server in VSCode*
- ~ *Formatting Tags*
- ~ *Article in HTML*
- ~ *Time & Address Tag*
- ~ *Quote & Cite*
- ~ *Strike*
- ~ *Progress Bar*
- ~ *Anchor Tag Styling*
- ~ *Image Tag*
- ~ *HTML Table*
- ~ *List*
- ~ *Input Tags,iframe*
- ~ *Forms*
- ~ *Video & Audio*
- ~ *iframe*
- ~ *Embed pdf*
- ~ *Maps*
- ~ *Symbols*
- ~ *Meta Tags*
- ~ *SVG*
- ~ *Emoji*

=> **CSS :**

- ~ *CSS Introduction*
- ~ *Inline vs Internal vs External*
- ~ *Priority between Inline, Internal & External*
- ~ *Multiple Properties in Single Element*
- ~ *Types of Selectors*
- ~ *Priority between Id, Class & Element*
- ~ *Comments*
- ~ *Colors*
- ~ *Backgroud*
- ~ *Border*
- ~ *Height & Width*
- ~ *Padding*
- ~ *Margin*
- ~ *Box Model*
- ~ *Text Properties*
- ~ *Anchor Tag Styling*
- ~ *Fonts*
- ~ *Cursor*
- ~ *!Important in CSS*
- ~ *Box Shadow*
- ~ *Opacity*
- ~ *Filter*
- ~ *Gradient*
- ~ *Overflow*
- ~ *List*
- ~ *Tables*
- ~ *Box Sizing*
- ~ *Inherit & Initial*
- ~ *Object Fit*
- ~ *Pseudo Classes*
- ~ *Pseudo Elements*
- ~ *Display*
- ~ *Position*
- ~ *Z-Index*
- ~ *Floats*
- ~ *2D Transform*
- ~ *Transitions*
- ~ *Flex*

- ~ Flex Direction & Wrap
- ~ Justify & Align in Flex
- ~ Order in Flex
- ~ Grow & Basis in Flex
- ~ Aling Items in Flex
- ~ Grids
- ~ Rows, Columns & Gap in Grids
- ~ Justify & Align in Grids
- ~ CSS Validator (Final Video)

=> Javascript :

- ~ Introduction
- ~ Running Javascript in Browser
- ~ Console
- ~ Strings & Numbers
- ~ var, let & const
- ~ Data Types
- ~ Type Conversions
- ~ Arithmetic Operators
- ~ Assignment Operator
- ~ Comparision Operator
- ~ Logir Not, Or and And
- ~ Swap Numbers
- ~ String Handling
- ~ String Searching
- ~ Arrays
- ~ Objects
- ~ Dates
- ~ Maths
- ~ If & Else
- ~ Challenge - If & Else
- ~ Switch Case
- ~ Challenge - Switch Case
- ~ JS Loops
- ~ For Loops
- ~ Nested Loops
- ~ Break & Continue
- ~ Arrays, Strings & Objects
- ~ For-in
- ~ For-of
- ~ While Loops
- ~ Do while Loops
- ~ Loops Exercies
- ~ Functions
- ~ Variable Scopes in Functions
- ~ Nested Functions
- ~ Parameters & Arguments
- ~ How function is useful?
- ~ Return in Function
- ~ Anonymous Functions
- ~ Calculator Exercise
- ~ Arrow Functions
- ~ forEach
- ~ maps
- ~ String Literals
- ~ Filter, Reduce & Every
- ~ Spread Operator
- ~ Challengege
- ~ Window & Document
- ~ Document Access
- ~ innerText & innerHTML
- ~ HTML Calculator
- ~ Query Selector
- ~ Styling in JS
- ~ Advance DOM Manipulation
- ~ Events
- ~ Basic Events
- ~ Time Events
- ~ Pop-up Boxes
- ~ Error Handling
- ~ Form Validation
- ~ Asynchronous JS
- ~ this keyword
- ~ useStrict
- ~ Hoisting
- ~ Local Storage
- ~ Session Storage
- ~ Cookies
- ~ Cookies vs Local Storage vs Session Storage
- ~ JSON vs Object literals
- ~ API
- ~ Fetching
- ~ Methods & Status Codes
- ~ Post Method
- ~ Put Method
- ~ Guess the Number
- ~ Generators
- ~ Regex

=> The JavaScript Standard Library :

- ~ *The JavaScript Standard*
- ~ *Sets and Maps*
- ~ *Typed Arrays and Binary Data*
- ~ *Pattern Matching with Regular Expressions*
- ~ *Dates and Times*
- ~ *Error Classes*
- ~ *JSON Serialization and Parsing*
- ~ *The Internationalization API*
- ~ *The Console API*
- ~ *URL APIs*
- ~ *Timers*

=> Iterators and Generators :

- ~ *What are Iterators and Generators?*
- ~ *How Iterators Work?*
- ~ *Implementing Iterable Objects*
- ~ *Generators*
- ~ *Advanced Generator Features*

=> Asynchronous JavaScript :

- ~ *What is Asynchronous JavaScript?*
- ~ *Asynchronous Programming with Callbacks*
- ~ *Promises*
- ~ *Async and await*
- ~ *Asynchronous Iteration*

=> Working with Web Browsers :

- ~ *JavaScript in Web Browsers*
- ~ *Web Programming Basics*
- ~ *Events*
- ~ *Scripting Documents*
- ~ *Scripting CSS*
- ~ *Document Geometry and Scrolling*
- ~ *Web Components*
- ~ *SVG: Scalable Vector Graphics*
- ~ *Audio APIs*
- ~ *Location, Navigation, and History*
- ~ *Networking Concepts*
- ~ *Storage*
- ~ *Worker Threads and Messaging*

=> Node.js :

- ~ *What is Node.js?*
- ~ *Client-Server Architecture*
- ~ *Single-Threaded Model*
- ~ *Multi-Threaded Model*
- ~ *Multi-Threaded vs Event-Driven*
- ~ *What is Node.js?*
- ~ *Node.js Features*
- ~ *Node.js Installation*
- ~ *Node.js First Example*
- ~ *Blocking vs Non-blocking*
- ~ *Global Objects*
- ~ *File System*
- ~ *Callbacks*
- ~ *Events*
- ~ *Node.js Architecture*
- ~ *NPM(Node Package Manager)*
- ~ *Node.js Modules*
- ~ *Node.js Modules Types*
- ~ *Core Modules*
- ~ *Local Modules*
- ~ *3rd Party Modules*
- ~ *JSON File*
- ~ *Variables*
- ~ *Operators*
- ~ *Functions*
- ~ *Objects*
- ~ *File Systems*
- ~ *Events*
- ~ *HTTP Module*
- ~ *Creating a Web Server using Node.js*
- ~ *Node.js NPM Tutorial*
- ~ *What is NPM?*
- ~ *Main Functions of NPM*
- ~ *Need For NPM*
- ~ *NPM Packages*
- ~ *NPM Installation*
- ~ *JSON File*
- ~ *Node.js Express Tutorial*
- ~ *Introduction to Express.js*
- ~ *Features of Express.js*
- ~ *Getting Started with Express.js*
- ~ *Routing Methods*
- ~ *Building RESTful API with Node.js*
- ~ *What is REST API?*
- ~ *Features of REST API*
- ~ *Principles of REST API*

- ~ *Methods of REST API*
- ~ *Building REST API with Node.js*
- ~ *Contact List MERN App*

=> React JS :

- ~ *Introduction to React*
- ~ *Why should you learn React?*
- ~ *Features of React*
- ~ *React applications*
- ~ *React App & JSX*
- ~ *Functional Components*
- ~ *Applying CSS Styles*
- ~ *Click Events*
- ~ *useState Hook*
- ~ *Lists & Keys*
- ~ *Props & Prop Drilling*
- ~ *Controlled Component Inputs*
- ~ *Project Challenge*
- ~ *useEffect Hook*
- ~ *JSON Server*
- ~ *Fetch API Data*
- ~ *CRUD Operations*
- ~ *Fetch Data Challenge*
- ~ *React Router*
- ~ *Router Hooks*
- ~ *Links*
- ~ *Flexbox Components*
- ~ *Axios API Requests*
- ~ *Custom React Hooks*
- ~ *Context API & useContext Hook*
- ~ *Build & Deploy Your React Apps*

=> Javascript Projects :

- ~ *Creating shopping cart app with User Interface*

=> Bitcoin Blockchain :

- ~ *History of currencies*
- ~ *Fiat currencies*
- ~ *Disadvantages of fiat currencies*
- ~ *Global financial system*
- ~ *How Central Banks work?*
- ~ *The 2008 Global Financial Crisis*
- ~ *Aftermath of 2008 recession*
- ~ *Creation of Bitcoin- A new decentralised digital currency*
- ~ *Bitcoin message hash implementation in Javascript*
- ~ *Immutable ledger practical implementation*
- ~ *Genesis block*
- ~ *Timestamp server*
- ~ *Merkel trees*
- ~ *Bitcoin as a State Transition System*
- ~ *Unspent Transaction outputs(UTXOs) Javascript implementation*
- ~ *Bitcoin whitepaper*
- ~ *What is a block?*
- ~ *Components of a Bitcoin block*
- ~ *Bitcoin Blockchain live implementation*
- ~ *Distributed Blockchain*
- ~ *Centralized vs Distributed Blockchain*
- ~ *consensus mechanism*
- ~ *Why do we need consensus mechanism in Blockchain networks?*
- ~ *Byzantine generals problem*
- ~ *Byzantine fault tolerance- A solution to Byzantine generals problem*
- ~ *BFT javascript implementation*
- ~ *Bitcoin nodes*
- ~ *Bitcoin miners*
- ~ *Blockchain mining operation*
- ~ *Mempool*
- ~ *Bitcoin difficulty adjustment*
- ~ *Bitcoin halving cycle*
- ~ *Competing chain problem*
- ~ *Maintaining immutability - Longest Chain rule*
- ~ *Block validation*
- ~ *consensus rules*
- ~ *Double Spend Validation*
- ~ *Transaction Input and Output Validation*
- ~ *Coinbase Transaction Reward Validation*
- ~ *Coinbase Maturity*
- ~ *Coinbase Transaction Block Height*
- ~ *Signature Check Counting*
- ~ *SigChecks*
- ~ *Mining incentive*
- ~ *Mining optimized hardware*
- ~ *CPU processing power*
- ~ *GPUs for mining*
- ~ *Application Specific Integrated Circuits(ASIC) miners*
- ~ *CPU vs GPU vs ASIC miners*
- ~ *Distributed peer to peer Blockchain live implementation*
- ~ *Distributed peer to peer Blockchain Javascript implementation*
- ~ *Token transaction live implementation using distributed peer to peer blockchain*
- ~ *Coinbase transaction live implementation using distributed peer to peer blockchain*

- ~ Token and Coinbase transaction Javascript implementation
- ~ Bitcoin public key and private key
- ~ Public key and private key generation
- ~ Bitcoin addresses
- ~ Bitcoin digital signatures
- ~ Signing a peer to peer message with private key- Javascript implementation
- ~ Verifying peer to peer message using public key and digital signature-implementation
- ~ Signing and verifying currency transaction- implementation
- ~ Complete Bitcoin Blockchain implementation with transaction signatures

=> Probable attacks in Bitcoin blockchain :

- ~ Sybil Attack
- ~ Race Attack
- ~ Finney Attack
- ~ Vector76 Attack
- ~ 51% Attack

=> Bitcoin Project :

- ~ Building a Blockchain using Javascript

=> Ethereum Blockchain :

- ~ Module overview
- ~ Understanding the drawbacks of Bitcoin blockchain
- ~ Lack of Turing-completeness
- ~ Value-blindness
- ~ Lack of state
- ~ Blockchain-blindness
- ~ Origin of Ethereum- The programmable currency
- ~ The Decentralized Applications revolution and modern state of blockchain systems
- ~ Decentralized Applications vs Centralized Applications
- ~ Ethereum Accounts overview
- ~ Contract Accounts(CA)
- ~ Externally Owned Accounts(EOA)
- ~ Fields in Ethereum accounts
- ~ Ethereum Account messages
- ~ Ethereum Account transactions
- ~ Ethereum Addresses
- ~ Units of Ether
- ~ Ether Gas
- ~ Computing total gas cost for Ethereum transactions
- ~ Ethereum gas price Javascript implementation
- ~ Ethereum as a State Transition Function
- ~ Ethereum Architecture
- ~ Ethereum Virtual Machine(EVM)
- ~ EVM nodes vs mining nodes
- ~ EVM Bytecode
- ~ EVM Instruction Set
- ~ EVM Opcode
- ~ EVM Storage
- ~ EVM Memory
- ~ EVM Stack
- ~ Geth setup and EVM practical
- ~ Converting bytecode to opcode
- ~ Application Binary Interface(ABI)
- ~ Understanding end-to-end Ethereum Blockchain transaction in Javascript
- ~ Ethereum Smart Contracts architecture

=> Ethereum 2.0 :

- ~ Why was Ethereum 2.0 proposed?
- ~ Energy usage in Proof of Work
- ~ Gas costs in Ethereum 1.0
- ~ Potential scalability issues
- ~ Moving from Proof of Work to Proof of Stake
- ~ Proof of Stake in Ethereum 2.0
- ~ Validators
- ~ Staking
- ~ Attestation
- ~ Crosslinks
- ~ Finality
- ~ consensus clients
- ~ Execution clients
- ~ Sharding
- ~ Shard chains
- ~ Beacon chain
- ~ Data rollup in Ethereum 2.0
- ~ Forking in Blockchain
- ~ Hard Fork
- ~ Soft Fork
- ~ The DAO attack and Ethereum Hard Fork

=> Solidity :

- ~ What is Solidity?
- ~ Why should you learn Solidity programming?
- ~ Introduction to Smart Contracts
- ~ Solidity Installation
- ~ Remix IDE
- ~ Installing Solidity in npm / Node.js
- ~ Layout of a Solidity Source File
- ~ SPDX License Identifier
- ~ Pragmas

- ~ *Comments in Solidity*
- ~ *Structure of a Smart Contract*

=> Solidity Value Types :

- ~ *Solidity datatypes*
- ~ *Booleans*
- ~ *Integers*
- ~ *Address Type*
- ~ *Address Literals*
- ~ *Contract Types*
- ~ *Byte Type*
- ~ *String Types*
- ~ *Enums in Solidity*

=> Solidity Reference Types :

- ~ *Data locations- storage, memory and callback*
- ~ *Solidity Arrays*
- ~ *Fixed Arrays*
- ~ *Dynamic Arrays*
- ~ *Bytes and Strings as Arrays*
- ~ *Array Slicing*
- ~ *Structs*
- ~ *Mapping Types*

=> Solidity Units and Global Variables :

- ~ *Ether Units*
- ~ *Time Units*

=> Solidity Control Structures :

- ~ *If statement*
- ~ *If/else statement*
- ~ *Nested if/else statements*
- ~ *Solidity Loops*
- ~ *For loop*
- ~ *While loop*
- ~ *Do-while loop*
- ~ *Break statement*
- ~ *Continue statement*

=> ABI Encoding and Decoding Functions :

- ~ *ABI encoder*
- ~ *ABI decoder*

=> Cryptographic Functions :

- ~ *Keccak256*
- ~ *SHA256*
- ~ *Ripemd160*
- ~ *Ecrecover*

=> Smart Contracts :

- ~ *Creating Smart Contracts*
- ~ *Constructor*
- ~ *Scope visibility*
- ~ *State variable visibility*
- ~ *Functions*
- ~ *Function visibility*
- ~ *Getter functions*
- ~ *Setter functions*
- ~ *Function modifiers*
- ~ *Return variables and returning multiple values*
- ~ *Immutable state variables*
- ~ *Payable functions*
- ~ *Fallback functions*
- ~ *View functions*
- ~ *Pure functions*
- ~ *Function overloading*
- ~ *Function overriding*
- ~ *Solidity Events*
- ~ *Block and Transaction details*
- ~ *Solidity Inheritance*
- ~ *Single Inheritance*
- ~ *Multiple Inheritance*
- ~ *Heirarchical Inheritance*
- ~ *Multilevel Inheritance*
- ~ *Abstract Contracts*
- ~ *Solidity Interfaces*
- ~ *Solidity Libraries*

=> Solidity Programming Applications :

- ~ *Ether Wallet*
- ~ *Multi Sig Wallet*
- ~ *Iterable Mapping*
- ~ *ERC20*
- ~ *ERC721*
- ~ *Uni-directional Payment Channel*
- ~ *Bi-directional Payment Channel*
- ~ *NFT Auction*
- ~ *Crowd Fund*
- ~ *Time Lock*

=> Common Ethereum Blockchain Hacks and Loopholes :

- ~ Re-Entrancy Attack
- ~ Self Destruct
- ~ Accessing Private Data
- ~ Denial of Service
- ~ Phishing with tx.origin
- ~ Hiding Malicious Code with External Contract
- ~ Honeypot
- ~ Front Running
- ~ Block Timestamp Manipulation
- ~ Signature Replay
- ~ Bypass Contract Size Check

=> Introduction to Blockchain Development Frameworks :

- ~ Introduction to Smart Contract Development in Production
- ~ Web3 libraries for Javascript
- ~ Smart Contract development tools
- ~ Web3 Providers
- ~ Wallets

=> Truffle Suite :

- ~ Truffle overview
- ~ Truffle Installation
- ~ Creatin a new project in Truffle
- ~ Exploring project directories in Truffle
- ~ Compiling Smart Contracts
- ~ Building Artifacts
- ~ Handling Dependencies
- ~ Reading and writing Smart Contract data
- ~ Smart Contract Transactions in Truffle
- ~ Function calls in Truffle
- ~ Abstractions
- ~ Executing Contract functions
- ~ Making Transactions
- ~ Processing Transaction results
- ~ Catching events
- ~ Add a new contract to the network
- ~ Sending ether to a contract
- ~ Invoking overloaded methods
- ~ Using enumerations
- ~ Preserving Files and Content to Storage Platforms
- ~ Inter Planetary File System(IPFS)
- ~ Filecoin
- ~ Textile Buckets
- ~ Running Migrations
- ~ Initial Migration
- ~ Truffle Deployer
- ~ Network considerations
- ~ Truffle Deployer API
- ~ Integrating Truffle with Metamask
- ~ Using Truffle Dashboard
- ~ Using truffle Debugger
- ~ Truffle Develop and Truffle Console
- ~ Writing and executing external scripts
- ~ Testing Smart Contracts
- ~ Writing Automated Tests in Javascript
- ~ Writing Automated Tests in Solidity
- ~ Truffle Build Process
- ~ Truffle Boxes
- ~ Ethereum Name Service
- ~ Truffle Event System
- ~ Network Configuration and Dapp Deployment
- ~ Ganache- Ethereum Client for Truffle Suite
- ~ Installing Ganache
- ~ Ganache Workspaces
- ~ Ganache Ethereum Workspace
- ~ Understanding Workspace Default Configuration in Ganache
- ~ Managing Ganache configurations and settings
- ~ Configuring Truffle to connect to Ganache
- ~ Managing Truffle projects in Ganache
- ~ Exploring the Contracts page
- ~ Exploring the Transactions page
- ~ Linking and unlinking a Truffle project
- ~ Ganache Workspaces
- ~ Creating Workspaces
- ~ Deleting Workspaces
- ~ Editing Workspaces
- ~ Ethereum Workspace
- ~ Loading Existing Workspaces
- ~ Switching Workspaces

=> Hardhat :

- ~ Introduction To Hardhat - Ethereum development environment for professionals
- ~ Hardhat Installation
- ~ Creating a Hardhat project
- ~ Configuring Ethereum Networks
- ~ Configuring the compiler
- ~ Compiling your contracts
- ~ Artifacts
- ~ Writing deployment scripts

- ~ Deploying the Contracts
- ~ Testing Smart Contracts
- ~ Running tests with Ganache
- ~ Running tests on Visual Studio Code
- ~ Running multiple tests in parallel
- ~ Running tasks
- ~ Hardhat Console
- ~ Creating custom tasks
- ~ Hardhat Runtime Environment(HRE)
- ~ Hardhat Plugins
- ~ Optimizing Plugins
- ~ Verbose Logging for debugging
- ~ Solutions to common runtime problems

=> Web3.js :

- ~ Introduction to Web3.js
- ~ Why should you learn Web3.js?
- ~ Applications of Web3.js
- ~ Installing Web3.js using NPM
- ~ Web3 modules
- ~ Creating a new Web3 instance
- ~ Introduction to Web3 Providers
- ~ Setting up a Web3 Provider
- ~ Batch request
- ~ Extending Web3 modules
- ~ Introduction to Web3.eth
- ~ Checksum addresses overview
- ~ Fetching default blockchain details
- ~ Transaction methods
- ~ Block Node methods
- ~ Subscriber Methods
- ~ Web3.js Smart Contract objects and methods
- ~ User wallet and account methods
- ~ Interacting with Ethereum node accounts using web3.eth.personal
- ~ Working with ABI in web3.js
- ~ Commonly used utilities in web3.js
- ~ Hardhat automated testing with Web3.js and Truffle

=> Ethers.js :

- ~ What is Ethers?
- ~ Ethers.js Features
- ~ Installing Ethers.js using NPM
- ~ Connecting to Ethereum: MetaMask
- ~ Connecting to Ethereum: RPC
- ~ Building blocks of Ethers.js- Signers, Providers and Contracts

=> Ethers.js Providers :

- ~ What are Providers?
- ~ Ethers.js provider API overview
- ~ Provider Account methods
- ~ Blocks Methods
- ~ Ethereum Naming Service (ENS) Methods
- ~ EnsResolver
- ~ Logs Methods
- ~ Network Status Methods
- ~ Transactions Methods
- ~ Event Emitter Methods
- ~ Inspection Methods
- ~ BaseProvider
- ~ JsonRpcProvider
- ~ JsonRpcSigner
- ~ JsonRpcUncheckedSigner
- ~ Static.JsonRpcProvider
- ~ Node-Specific Methods
- ~ API Providers
- ~ EtherscanProvider
- ~ InfuraProvider
- ~ AlchemyProvider
- ~ CloudflareProvider
- ~ PocketProvider
- ~ AnkrProvider
- ~ Other Providers
- ~ FallbackProvider
- ~ IpcProvider
- ~ JsonRpcBatchProvider
- ~ UrlJsonRpcProvider
- ~ Web3Provider
- ~ WebSocketProvider

=> Smart Contract Interaction :

- ~ Creating new Smart Contract instance
- ~ Contract Properties
- ~ Contract Methods
- ~ Events
- ~ ContractFactory
- ~ Creating ContractFactory Instances
- ~ ContractFactory Interface Properties
- ~ ContractFactory Methods
- ~ Meta-Class
- ~ Deploying a Contract

- ~ *Connecting to a Contract*

=> Ethereum Blockchain Projects :

- ~ *Building cryptocurrency with ICO*
- ~ *Building decentralized ecommerce website*
- ~ *Building decentralized voting application*
- ~ *Decentralized music sharing app*
- ~ *Token contract swap application*
- ~ *Full stack email dapp*

=> Oracles :

- ~ *What is a Blockchain Oracle?*
- ~ *Solving the Oracle problem*
- ~ *Decentralized Oracles*
- ~ *Types of Blockchain Oracles*
- ~ *Applications of Blockchain oracles*

=> Chainlink overview :

- ~ *Introduction to Chainlink*
- ~ *Understanding the Chainlink Ecosystem*
- ~ *Chainlink Features*
- ~ *Chainlink Applications as Decentralized Oracles*

=> Data Feeds :

- ~ *Introduction to Data Feeds*
- ~ *Using Data Feeds*
- ~ *Fetchin Historical Cryptocurrency Price Data*
- ~ *Chainlink Feed Registry*
- ~ *Using ENS with Data Feeds*

=> Custom Data Feeds :

- ~ *Using any API*
- ~ *Make a GET Request*
- ~ *Multi-Variable Responses*
- ~ *Large Responses*
- ~ *Make an Existing Job Request*
- ~ *Find Existing Jobs*
- ~ *Contract Addresses*

=> Oracle Projects :

- ~ *Live cryptocurrency trading using chainlink*
- ~ *Insurance Dapp using chainlink*

=> The Graph :

- ~ *The Graph Protocol*
- ~ *The Graph architecture*
- ~ *Edge and Node*
- ~ *Everest Registry*
- ~ *Graph Protocol*
- ~ *The Graph vs Etherscan*
- ~ *Graph-cli Installation*
- ~ *Creating new subgraphs*
- ~ *Writing subgraphs*
- ~ *Publishing a Subgraph to the Decentralized Network*

=> GraphQL API :

- ~ *Queries*
- ~ *Sorting*
- ~ *Pagination*
- ~ *Filtering*
- ~ *Time-travel queries*
- ~ *Fulltext Search Queries*
- ~ *Validation*
- ~ *Schema*
- ~ *Entities*
- ~ *Signalling*
- ~ *Curation*
- ~ *Delegators*
- ~ *Consumers*
- ~ *Deploying subgraphs*
- ~ *Subgraph logging*
- ~ *Graph protocol testnet using docker compose*
- ~ *Ethereum node monitoring using The Graph, Prometheus and Grafana*

=> The Graph Networking :

- ~ *Introduction to indexers*
- ~ *Revenue streams*
- ~ *Distribution*
- ~ *Allocation life cycles*
- ~ *Querying and indexing subgraphs*
- ~ *IPFS Hash convertor*

=> AssemblyScript API for The Graph :

- ~ *Installing AssemblyScript API*
- ~ *API Reference*
- ~ *Versions*
- ~ *Built-in Types*
- ~ *Store API*
- ~ *Ethereum API*
- ~ *Logging API*
- ~ *IPFS API*
- ~ *Crypto API*

- ~ JSON API
- ~ Type Conversions Reference
- ~ Data Source Metadata
- ~ Entity and DataSourceContext

=> Project :

- ~ Building a Full-stack Blockchain Application using Ethereum, Polygon, Next.js and GraphQL

=> Decentralized Autonomous Organisations(DAO) :

- ~ What are DAOs?
- ~ Why do we need DAOs?
- ~ DAO membership
- ~ Token-based membership
- ~ Share-based membership
- ~ How do DAOs work?
- ~ Properties of DAOs
- ~ Ethereum and DAOs
- ~ Understanding Governance Mechanisms
- ~ DAOs and the principal-agent problem
- ~ Building Decentralized Autonomous Organisations
- ~ Defining the DAO purpose
- ~ Building the DAO voting mechanism
- ~ Creating the governance token
- ~ DAO fund management
- ~ Initial Coin Offering (ICO)
- ~ Creating a DAO on Aragon
- ~ Creating a DAO using Snapshot
- ~ Building a DAO using DAOstack Alchemy

=> Creating a Custom DAO Project :

- ~ Understanding custom DAOs
- ~ Finding the purpose for our Custom DAO
- ~ Designing the voting architecture
- ~ Implementing the voting architecture in Solidity
- ~ Designing the components of the governance token(DAO cryptocurrency)
- ~ Creating the governance token in Solidity
- ~ Fund Management for our custom DAO
- ~ Designing the Multi-signature wallet for Fund Management
- ~ Creating the Multi-signature wallet in Solidity
- ~ Testing DAO Smart Contracts
- ~ Deploying the DAO to testnet

=> NFT Platforms :

- ~ What are NFT Platforms/Marketplaces?
- ~ CryptoKitties
- ~ Opensea
- ~ Rarible
- ~ Decentraland
- ~ Binance NFT
- ~ Enjin Marketplace
- ~ Axie Marketplace
- ~ Foundation
- ~ Nifty Gateway
- ~ Mintable
- ~ Theta Drop

=> NFT Transaction Fees :

- ~ Gas Fees in NFT
- ~ What are one-time Gas Fees NFT?
- ~ Recurring Gas Fees
- ~ Actions in Gas Fees
- ~ Check Ethereum Gas Fee
- ~ Create and Sell NFTs without Gas Fees
- ~ NFT Marketplace Fees

=> NFT project :

- ~ Building a complete NFT Marketplace with User Interface

=> Polygon Blockchain(MATIC) :

- ~ Introduction to Polygon Blockchain
- ~ Why should you use Polygon network?
- ~ Layer 1 vs Layer 2 Blockchains
- ~ Features of Polygon Blockchain
- ~ Polygon Architecture
- ~ Zero-Knowledge cryptography
- ~ Zero-Knowledge rollups

=> Polygon Projects :

- ~ Retail supply chain Application using Polygon Network
- ~ Building a Social media Dapp on Polygon

=> Polkadot :

- ~ Polkadot Overview
- ~ Polkadot Whitepaper
- ~ Polkadot Architecture
- ~ Parachains
- ~ Parathreads
- ~ Substrate Installation

=> Substrate Fundamentals :

- ~ Runtime environment and setup
- ~ Exinsics

- ~ *Account Abstractions*
- ~ *Transaction Pool*
- ~ *Session Keys*
- ~ *Transaction Weight*
- ~ *Execution*
- ~ *Off-Chain Features*

=> Runtime Development :

- ~ *Frames*
- ~ *Macros*
- ~ *Metadata*
- ~ *Storage*
- ~ *Origins*
- ~ *Events and Errors*
- ~ *Weights and Fees*
- ~ *Benchmarking*
- ~ *Debugging*
- ~ *Testing*
- ~ *Randomness*
- ~ *Chain Specification*
- ~ *Upgrades*
- ~ *Pallet Coupling*
- ~ *Custom RPCs*
- ~ *Smart Contract Toolkits*

=> Development Integration :

- ~ *Polkadot-JS*
- ~ *Client Libraries*
- ~ *Substrate Connect*

=> Development Tools :

- ~ *SR tool*
- ~ *Subxt*
- ~ *Tx Wrapper*
- ~ *Sub Flood*
- ~ *Substrate Archive*
- ~ *Sidecar*
- ~ *Polkadot Launch*

=> Advanced topics in Polkadot :

- ~ *Account Info*
- ~ *SCALE Codec for Substrate*
- ~ *Consensus*
- ~ *Block Import*
- ~ *Executor*
- ~ *Cryptography*
- ~ *Storage*
- ~ *SS58 Address Format*
- ~ *Hash Collections*

Dart Programming

Topic Name : MOBILE DEVELOPEMENT

Sub-topic Name : DART

Course link : <https://ineuron.ai/course/Dart-Programming>

Course Description :-

Learn how to write Dart programmes from the ground up. This course is designed for those who have never programmed before. Dart is a strong and expressive language with a simple learning curve. This makes it an excellent first language. Dart provides a client-optimized language, rich and powerful frameworks, and flexible tools to help you create attractive, high-quality experiences across all screens.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => Dart fundamentals
- => Data types in Dart
- => Arrays
- => Maps
- => Constants
- => Operators
- => Conditionals
- => Functions
- => Object Oriented Programming in Dart
- => Asynchronous subroutines

Requirements :-

- => System with minimum i3 processor or better
- => At least 4 GB of RAM
- => Working internet connection
- => Dedication to learn

Instructors :-

=> Hitesh Choudhary :

~ I like to make videos related to code and tech in my free time. I also lead a few tech teams in startups, help in hiring talent for companies. I am also on a part time traveller, with 31 countries checked off so far!

Curriculum details :-

=> Introduction to Dart programming language :

~ Introduction to dart

=> Hello World on MAC in Dart :

~ Dart installation on MAC

=> Hello World on Windows in Dart :

~ Dart installation for WINDOWS

=> Basics of Dart :

~ Introduction to variables

~ datatypes in dart

~ array in dart

~ Maps in dart

~ Constants and operations

=> Conditionals in Dart :

~ Introduction to if and else in dart

~ Advance if else statement in dart

~ Switch and case in dart

=> Functions and loops in Dart :

- ~ *basics of functions*
- ~ *Creating a calculator in dart*
- ~ *advance functions in dart*
- ~ *While loop in dart*
- ~ *Do While loop in dart*
- ~ *Solution for assignment in dart*
- ~ *for and for in loop in dart*

=> Intermediate Dart :

- ~ *More on Arrays in dart*
- ~ *More on maps in dart*
- ~ *for each loop for map in dart*

=> OOPS in Dart :

- ~ *Classes in dart*
- ~ *Objects in dart*
- ~ *constructor in Dart*
- ~ *getters and setters in dart*
- ~ *inheritance in dart*
- ~ *Interface in dart*
- ~ *Using multiple files in classes*

=> Advance Dart :

- ~ *Generics in dart*
- ~ *Code cascading in dart*
- ~ *Custom exception handling*
- ~ *Dart libraries*
- ~ *future and async in dart*
- ~ *Web server in dart*
- ~ *Making an API request in dart*

AWS Interview Preparation

Topic Name : CLOUD

Sub-topic Name : AWS

Course link : <https://ineuron.ai/course/AWS-Interview-Preparation>

Course Description :-

Amazon Web Service (AWS) is one of the fastest-growing fields in the technology world. This course is designed to help you achieve your goals which cover interview questions. We cover a wide range of topics in this course. We have questions on Amazon Web Services (AWS) best practices, Security, Simple Storage Service S3, Elastic Compute Cloud EC2, Elastic Load Balancing ELB, CloudFront, DynamoDB, CloudWatch, ElastiCache and Lambda.

Course Features :-

- => Self-Paced Recording
- => Assignment in all modules
- => Quiz in every module
- => Completion Certificate

What you will learn :-

- => Aws interview questions
- => Learn about IAM
- => Learn about EC2

Requirements :-

- => Understanding of AWS
- => A system with internet connection
- => Your dedication
- => Interest to learn

Instructors :-

- => MD Imran :
 - ~ Working as Data Scientist with experience in solving real world business problems across different domains.

Curriculum details :-

=> The Cloud :

- ~ Which of the following does not contribute significantly to the operational value of a large cloud provider like AWS?
- ~ Which of the following are signs of a highly available application? (Select TWO.)
- ~ How does the metered payment model make many benefits of cloud computing possible? (Select TWO.)
- ~ Which of the following are direct benefits of server virtualization? (Select TWO.)
- ~ What is a hypervisor? Preview
- ~ Which of the following best describes server virtualization?
- ~ Which of the following best describes Infrastructure as a Service products?
- ~ Which of the following best describes Platform as a Service products?
- ~ Which of the following best describes Software as a Service products?
- ~ Which of the following best describes scalability?
- ~ Which of the following best describes elasticity?
- ~ Which of the following characteristics most help AWS provide such scalable services? (Select TWO.)

=> Understanding Your AWS Account :

- ~ Which of the following EC2 services can be used without charge under the Free Tier?
- ~ You want to experiment with deploying a web server on an EC2 instance. Which two of the following resources can you include to make that work while remaining within the Free Tier? (Select TWO.)
- ~ Which of the following usage will always be cost-free even after your accounts Free Tier has expired? (Select TWO.)
- ~ Which of the following tools are available to ensure you won't accidentally run past your Free Tier limit and incur unwanted costs? (Select TWO.)
- ~ Which of the following is likely to be an accurate source of AWS pricing information?
- ~ Which of the following will probably not affect the pricing for an AWS service?
- ~ Which of the following is a limitation of the AWS Simple Monthly Calculator?
- ~ Which of the following Simple Monthly Calculator selections will likely have an impact on most other configuration choices on the page? (Select TWO.)
- ~ Which of the following is not an included parameter in the AWS Total Cost of Ownership Calculator?
- ~ Which of the following AWS Total Cost of Ownership Calculator parameters is likely to have the greatest impact on cost?
- ~ Which of the following AWS documentation URLs points to the page containing an up-to-date list of service limits?
- ~ Which of the following best describes one possible reason for AWS service limits?
- ~ Is it always possible to request service limit increases from AWS? Preview
- ~ Which is the best place to get a quick summary of this month's spend for your account?
- ~ What is the main goal for creating a Usage budget type (in AWS Budgets)?
- ~ Which of the following is not a setting you can configure in a Cost budget?
- ~ What is the main difference between the goals of Cost Explorer and of cost and usage reports?
- ~ What is the purpose of cost allocation tags?
- ~ Which of the following scenarios would be a good use case for AWS Organizations? (Select TWO.)
- ~ Which of these tools lets you design graphs within the browser interface to track your account spending?

=> Getting Support on AWS :

- ~ Your company is planning a major deployment on AWS. While the design and testing stages are still in progress, which of the following plans will provide the best blend of support and cost savings?
- ~ Your web development team is actively gearing up for deployment of an e-commerce site. During these early stages of the process, individual developers are running into frustrating conflicts and configuration problems that are highly specific to your situation. Which of the following plans will provide the best blend of support and cost savings?
- ~ Your corporate website was offline last week for more than two hours which caused serious consequences, including the early retirement of your CTO. Your engineers have been having a lot of trouble tracking down the source of the outage and admit that they need outside help. Which of the following will most likely meet that need?
- ~ For which of the following will AWS provide direct 24/7 support to all users even those on the Basic Support plan?
- ~ The primary purpose of an AWS technical account manager is to:
- ~ Your Linux-based EC2 instance requires a patch to a Linux kernel module. The problem is that patching the module will, for some reason, break the connection between your instance and data in an S3 bucket. Your team doesn't know if it's possible to work around this problem. Which is the most cost-effective AWS plan through which support professionals will try to help you?
- ~ Your company enrolled in the Developer Support plan and, through the course of one month, consumed \$4,000 USD of AWS services. How much will the support plan cost the company for the month?
- ~ Your company enrolled in the Business Support plan and, through the course of three months, consumed \$33,000 of AWS services (the consumption was equally divided across the months). How much will the support plan cost the company for the full three months?
- ~ Which of the following AWS support services does not offer free documentation of some sort?
- ~ What is the key difference between the roles of AWS Professional Services and a technical account manager (TAM)?
- ~ AWS documentation is available in a number of formats, including which of the following? (Select TWO.)
- ~ Which of the following documentation sites are most likely to contain code snippets for you to cut and (after making sure you understand exactly what they'll do) paste into your AWS operations? (Select TWO.)
- ~ What is the primary function of the content linked from the Knowledge Center?
- ~ On which of the following sites are you most likely to find information about encrypting your AWS resources?
- ~ When using AWS documentation pages, what is the best way to be sure the information you're reading is up-to-date?
- ~ Which of the following is not a Trusted Advisor category?
- ~ Data volumes that aren't properly backed up is an example of which of these Trusted Advisor categories?
- ~ Instances that are running (mostly) idle should be identified by which of these Trusted Advisor categories?
- ~ Within the context of Trusted Advisor, what is a false positive?
- ~ Which of the following Trusted Advisor alerts is available only for accounts on the Business or Enterprise Support plan? (Select TWO.)

=> Understanding the AWS Environment :

- ~ Which of the following designations would refer to the AWS US West (Oregon) region?
- ~ Which of the following is an AWS Region for which customer access is restricted?
- ~ When you request a new virtual machine instance in EC2, your instance will automatically launch into the currently selected value of which of the following?
- ~ Which of the following are not globally based AWS services? (Select TWO.)
- ~ Which of the following would be a valid endpoint your developers could use to access a particular Relational Database Service instance you're running in the Northern Virginia region?
- ~ What are the most significant architectural benefits of the way AWS designed its regions? (Select TWO.)
- ~ Why is it that most AWS resources are tied to a single region?
- ~ You want to improve the resilience of your EC2 web server. Which of the following is the most effective and efficient approach?
- ~ Which of the following is the most accurate description of an AWS Availability Zone?
- ~ Which of the following most accurately describes a subnet within the AWS ecosystem?
- ~ What determines the order by which subnets/AZ options are displayed in EC2 configuration dialogs?
- ~ What is the primary goal of autoscaling? Preview
- ~ Which of the following design strategies is most effective for maintaining the reliability of a cloud application?
- ~ Which of the following AWS services are not likely to benefit from Amazon edge locations? (Select TWO.)
- ~ Which of the following is the primary benefit of using CloudFront distributions?
- ~ What is the main purpose of Amazon Route 53?
- ~ According to the AWS Shared Responsibility Model, which of the following are responsibilities of AWS? (Select TWO.)
- ~ According to the AWS Shared Responsibility Model, what's the best way to define the status of the software driving an AWS managed service?
- ~ Which of the following is one of the first places you should look when troubleshooting a failing application?
- ~ Where will you find information on the limits AWS imposes on the ways you can use your account resources?

=> Securing Your AWS Resources :

- ~ What is the primary function of the AWS IAM service?
- ~ Which of the following are requirements you can include in an IAM password policy? (Select THREE.)
- ~ Which of the following should you do to secure your AWS root user? (Select TWO.)
- ~ How does multi-factor authentication work?
- ~ Which of the following SSH commands will successfully connect to an EC2 Amazon Linux instance with an IP address of 54.7.35.103 using a key named mykey.pem?
- ~ What's the most efficient method for managing permissions for multiple IAM users?
- ~ What is an IAM role?
- ~ How can federated identities be incorporated into AWS workflows? (Select TWO.)
- ~ Which of the following are valid third-party federated identity standards? (Select TWO.)
- ~ What information does the IAM credential report provide?
- ~ What text format does the credential report use?
- ~ Which of the following IAM policies is the best choice for the admin user you create in order to replace the root user for day-to-day administration tasks?
- ~ What will you need to provide for a new IAM user you're creating who will use programmatic access to AWS resources?
- ~ What will IAM users with AWS Management Console access need to successfully log in?
- ~ Which of the following will encrypt your data while in transit between your office and Amazon S3?
- ~ Which of the following AWS resources cannot be encrypted using KMS?
- ~ What does KMS use to encrypt objects stored on your AWS account?
- ~ Which of the following standards governs AWS-based applications processing credit card transactions?
- ~ What is the purpose of the Service Organization Controls (SOC) reports found on AWS Artifact?
- ~ What role can the documents provided by AWS Artifact play in your application planning? (Select TWO.) Preview

=> Working with Your AWS Resources :

- ~ Which of the following credentials can you use to log into the AWS Management Console?
- ~ How long will your session with the AWS Management Console remain active?
- ~ While looking at the EC2 service console in the AWS Management Console while logged in as the root user, you notice all of your instances are missing. What could be the reason?
- ~ Which of the following is true regarding a resource tag?
- ~ Which of the following is required to use the AWS Command Line Interface (CLI)?
- ~ Which of the following are options for installing the AWS CLI on Windows 10? (Select TWO.)
- ~ After installing the AWS Command Line Interface, what should you do before using it to securely manage your AWS resources?

~ Which output format does the AWS CLI support?

Full Stack Web Development with Python in Hindi Tech Neuron

Topic Name : DATA SCIENCE

Sub-topic Name : PYTHON

Course link : <https://ineuron.ai/course/Full-Stack-Web-Development-with-Python-in-Hindi-Tech-Neuron>

Course Description :-

Full Stack Python with Django is specially created to fulfil the standards set by the industry. You may learn a lot about Python, the Django REST framework, Django Models, React, and other topics in this lengthy online interactive course. With the Python Django program, you will also gain proficiency with the web framework while working on practical use cases.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate
- => 18 hr live support all seven day
- => Extra doubt clearing as per demand
- => hindi
- => Doubt clearing
- => 300+ practice problems

What you will learn :-

- => Python basic
- => List object basics
- => String objects
- => Tuples
- => Functions
- => Memory management
- => OOPS concepts
- => Exception Handling
- => Files
- => DATABASE
- => Web Development using Python on Django
- => React
- => Web API

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Instructors :-

=> Saurabh Shukla :

~ Saurabh Shukla has been educating children with this credo, and he does so for free. MySirG.com, the educator's YouTube channel, features video lessons on programming languages. Saurabh has successfully reached thousands of students around the country by making it his aim to produce subject-related videos on a daily basis.

Curriculum details :-

- => Course introduction :
 - ~ course overview and dashboard description
 - ~ Programming language overview
 - ~ History of Python
 - ~ Version History of Python
 - ~ Installation (tools:vscode, IDLE)
 - ~ Learning Path through Python

=> Python basic :

- ~ *Developing and executing Python program*
- ~ *Python Virtual Machine*
- ~ *Comment, data, variables, types*
- ~ *heap space and namespace*
- ~ *id, keywords, import, module, help*
- ~ *print, input*
- ~ *number system*
- ~ *Conversion functions*
- ~ *Operators - arithmetic, bitwise, comparison and assignment operators, operators precedence and associativity*
- ~ *Decision Control Statements*
- ~ *Iterative control Statements - while loop*
- ~ *transfer control statements- break, continue, pass*
- ~ *for loop*
- ~ *range*

=> List object basics :

- ~ *list*
- ~ *list methods*
- ~ *List comprehensions*
- ~ *packing unpacking*
- ~ *built-in methods*

=> String objects :

- ~ *String object basics*
- ~ *String methods*
- ~ *Splitting and joining strings*
- ~ *String format functions*

=> Tuples :

- ~ *tuple basics*
- ~ *tuple methods*
- ~ *list vs tuple*

=> set :

- ~ *set basics*
- ~ *set comprehensions*
- ~ *set methods*

=> dict :

- ~ *dictionary basics*
- ~ *dict methods*
- ~ *dict comprehensions*

=> Functions :

- ~ *Function Basics*
- ~ *Ways to define a function*
- ~ *function returning None*
- ~ *default arguments*
- ~ *positional vs keyword arguments*
- ~ *variable number of arguments*
- ~ *Recursion*
- ~ *Lambda functions*
- ~ *Iterators*
- ~ *Generator functions*
- ~ *Decorators*
- ~ *Map, reduce, filter functions.*

=> Memory management :

- ~ *Multithreading*
- ~ *Multiprocessing*

=> OOps concepts :

- ~ *oops basic concepts.*
- ~ *Main aspects of OOP*
- ~ *Encapsulation and Abstraction*
- ~ *Classes and Objects*
- ~ *init method*
- ~ *Types of variables*
- ~ *Types of functions*
- ~ *Inheritance*
- ~ *Name conflict issues*
- ~ *Polymorphism*
- ~ *Overriding*
- ~ *Operator Overloading*

=> Exception Handling :

- ~ *Introduction to Exceptions*
- ~ *Exceptions handling with try-except*
- ~ *use of else and finally*
- ~ *Defining exception*

=> Files :

- ~ *Working with files*
- ~ *Reading and writing files*
- ~ *Buffered read and write*
- ~ *Other file methods.*
- ~ *Renaming a file*
- ~ *Removing a file*

=> DATABASE :

~ *Postgres Basics*

=> **Project :**

~ *Core Python + DB*

~ *Web Project-1*

~ *Web Project-2*

=> **Web Development using Python on Django :**

~ *Client Server Architecture*

~ *HTML*

~ *CSS*

~ *JS*

~ *Introduction to Django*

~ *Creating first Django Project*

~ *Creating web application*

~ *Defining views in an application*

~ *Multiple views*

~ *Multiple Application*

~ *Application level url configuration*

~ *Template*

~ *Template variables*

~ *Template Tags*

~ *Template Filters*

~ *Custom Template Tags and Filters*

~ *Database Introduction*

~ *SQLite DB*

~ *Models*

~ *makemigrations*

~ *migrate*

~ *Admin app*

~ *CRUD operations*

~ *session tracking*

~ *login functionality*

=> **React :**

~ *React Essentials*

=> **Web API :**

~ *What is web API*

~ *Difference b/w API and web API*

~ *Restful services*

~ *POSTMAN*

Class 9th Biology

Topic Name : K12

Sub-topic Name : CLASS9

Course link : <https://ineuron.ai/course/Class-9th-Biology>

Course Description :-

The Science Syllabus is elegantly designed such that it introduces the basic concepts of science and its importance in our daily life.

Class 9th is crucial and is the foundation for higher education of students. The Biology section focuses on concepts like Cell - the fundamental unit of life, Diversity, Tissues, Natural Resources, etc.

Course Features :-

=> Self Paced Videos

=> Completion Certificate

What you will learn :-

=> Cell - the fundamental unit of life

=> Tissues

=> Diversity in living organisms

=> Why do we fall ill?

=> Natural Resources

=> Improvement in food resources

Requirements :-

=> System with Internet Connection

=> Interest to learn

=> Dedication

Instructors :-

=> Dr Nishtha Jain :

~ I am a doctor by profession but a teacher by passion. I have been into the teaching profession for the last 3 years. I have been and am still a mentor for various courses which include technical as well as non-technical ones. These include MS-Excel, Tableau, Computer basics, Biology, English, etc. I love to learn, explore and share my knowledge to whatever extent possible. Being an ardent educator, I have always helped all my students and will continue to do the same.

Curriculum details :-

=> Cell - the fundamental of life :

- ~ Lecture 1 : Cell Theory, Plasma Membrane Preview
- ~ Lecture 2 : Cell Wall, Nucleus, Prokaryotes VS Eukaryotes, Cytoplasm Preview
- ~ Lecture 3 : Cell organelles, ER, Golgi apparatus, Lysosomes Preview
- ~ Lecture 4 : Mitochondria, Plastids, Vacuoles, Cell division
- ~ Lecture 1 - NCERT Solutions
- ~ Lecture 2 - NCERT Solutions
- ~ Lecture 3 - NCERT Solutions

=> Tissues :

- ~ Lecture 1 : Introduction, Difference between Plant and Animal Tissues
- ~ Lecture 2 : Plant tissues and types (Meristematic)
- ~ Lecture 3 : Plant tissues and types (Permanent) , Epidermal & Guard cells
- ~ Lecture 4 : Plant tissues and types (Permanent), Xylem VS Phloem
- ~ Lecture 5 : Animal tissues and types (Epithelial)
- ~ Lecture 6 : Animal tissues and types (Connective)
- ~ Lecture 7 : Animal tissues and types (Muscular and Nervous)
- ~ Lecture 1 - NCERT Solutions
- ~ Lecture 2 - NCERT Solutions
- ~ Lecture 3 - NCERT Solutions
- ~ Lecture 4 - NCERT Solutions
- ~ Lecture 5 - NCERT Solutions

=> Diversity in living organisms :

- ~ Lecture 1 : Introduction, Hierarchy of Classification, Ranks
- ~ Lecture 2 : 5-Kingdom Classification, 6-Kingdom Classification
- ~ Lecture 3 : Whittaker's Classification (Monera, Protista)
- ~ Lecture 4 : Whittaker's Classification (Fungi, Plantae, Animalia)
- ~ Lecture 5 : Kingdom Plantae
- ~ Lecture 6 : Kingdom Animalia (1)
- ~ Lecture 7 : Kingdom Animalia (2)
- ~ Lecture 8 : Subphylum Vertebrata (1)
- ~ Lecture 9 : Subphylum Vertebrata (2)
- ~ Lecture 10 : Quick Notes, Nomenclature

=> Why do we fall ill :

- ~ Lecture 1 : *Healthy and Disease-free*
- ~ Lecture 2 : *Acute, Chronic, Infectious and Non-infectious; Causes*
- ~ Lecture 3 : *Spread of infectious diseases*
- ~ Lecture 4 : *Disease manifestations, Symptoms, Principles of treatment*
- ~ Lecture 5 : *Principles of disease prevention*
- ~ Lecture 1 - NCERT Solutions
- ~ Lecture 2 - NCERT Solutions
- ~ Lecture 3 - NCERT Solutions
- ~ Lecture 4 - NCERT Solutions

=> Natural Resources :

- ~ Lecture 1 : *Introduction*
- ~ Lecture 2 : *Convection*
- ~ Lecture 3 : *Condensation*
- ~ Lecture 4 : *Air Pollution*
- ~ Lecture 5 : *Water*
- ~ Lecture 6 : *Water Pollution*
- ~ Lecture 7 : *Soil*
- ~ Lecture 8 : *Soil layers, Soil Pollution, Ways to reduce soil pollution*
- ~ Lecture 9 : *Water Cycle*
- ~ Lecture 10 : *Nitrogen Cycle*
- ~ Lecture 11 : *Nitrogen Cycle Figure*
- ~ Lecture 12 : *Carbon Cycle and Greenhouse effect*
- ~ Lecture 13 : *Oxygen Cycle and Ozone Layer*
- ~ Lecture 14 : *Phosphorus Cycle*
- ~ Lecture 15 : *Sulfur Cycle*
- ~ Lecture 1 - NCERT Solutions
- ~ Lecture 2 - NCERT Solutions
- ~ Lecture 3 - NCERT Solutions
- ~ Lecture 4 - NCERT Solutions
- ~ Lecture 5 - NCERT Solutions
- ~ Lecture 6 - NCERT Solutions

=> Improvement in food resources :

- ~ Lecture 1 : *Introduction, Types of Revolutions*
- ~ Lecture 2 : *3 types of crops, Improvements in crop yields*
- ~ Lecture 3 : *Crop Variety Improvement*
- ~ Lecture 4 : *Crop Production Management (1)*
- ~ Lecture 5 : *Crop Production Management (2)*
- ~ Lecture 6 : *Crop Protection Management*
- ~ Lecture 7 : *Animal Husbandry - Cattle and Poultry Farming*
- ~ Lecture 8 : *Animal Husbandry - Fish farming and Bee keeping*
- ~ Lecture 1 - NCERT Solutions
- ~ Lecture 2 - NCERT Solutions
- ~ Lecture 3 - NCERT Solutions
- ~ Lecture 4 - NCERT Solutions
- ~ Lecture 5 - NCERT Solutions
- ~ Lecture 6 - NCERT Solutions
- ~ Lecture 7 - NCERT Solutions
- ~ Lecture 8 - NCERT Solutions

GCP Projects

Topic Name : CLOUD

Sub-topic Name : GCP PROJECT

Course link : <https://ineuron.ai/course/GCP-Projects>

Course Description :-

GCP is one of the most used and fastest-growing cloud platforms in the cloud industry currently. This course will get hands-on experience in building and implementing various real-time ML, DL & NLP-based projects with GCP cloud services.

Course Features :-

- => Real-time project implementation
- => Quizzes
- => Assignment problems
- => Downloadable resources
- => Completion certificate

What you will learn :-

- => GCP IAM and Security
- => GCP Storage & Database Services
- => GCP Management Tools
- => GCP AI Stack
- => Real-time Project implementation on ML, DL & NLP

Requirements :-

- => Prior Understanding of GCP
- => GCP Account
- => Prior understanding of ML, DL and NLP
- => A System with a decent internet connection
- => Your dedication

Instructors :-

=> Khushali Shah :

~ A data scientist having rich experience working with MNCs and start-ups in the field of data science and machine learning. She has expertise in Chatbot development for various domains & been developing professionally for 6+ years with diverse job history. She also had positions in software module development, web app development, functional designs, requirement gathering, client interaction, and server setup/admin & can help everywhere in the stack; she loves wearing multiple hats to an extent. She also believes in enhancing her skills by training and learning new things day by day.

Curriculum details :-

=> Projects :

- ~ Guest Book Project Preview
- ~ Overview Preview
- ~ Vertex AI image classification
- ~ Custom image classification
- ~ Install SDK
- ~ authenticating user
- ~ training
- ~ user_message

Big Data Masters Tech Neuron

Topic Name : BIG DATA

Sub-topic Name : BIG DATA MASTERS

Course link : <https://ineuron.ai/course/Big-Data-Masters-Tech-Neuron>

Course Description :-

This unique industry program will help to learn the entire stack of Big Data and be ready to crack jobs in leading organizations.

Course Features :-

- => Full stack Data Science masters certification
- => One year of internship Anytime
- => Online Instructor-led learning: Live teaching by instructors
- => 20 + hands-on industry real-time projects.
- => 200 hours live interactive classes.
- => Lifetime Dashboard access
- => Assignment in all the module
- => Quiz in every module
- => A live project with real-time implementation
- => Interview Preparation Anytime
- => Regular assessment

What you will learn :-

- => Big Data
- => Hadoop
- => HDFS
- => YARN
- => Linux
- => AWS EC2
- => AWS IAM
- => AWS S3
- => AWS SNS
- => AWS DMS
- => AWS RDS
- => AWS Redshift
- => Hbase
- => Sqoop
- => Confluent
- => Atlas
- => Ambari
- => Databricks

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Instructors :-

=> Sudhanshu Kumar :

~ Having 8+ years of experience in Big data, Data Science and Analytics with product architecture design and delivery. Worked in various product and service based Company. Having an experience of 5+ years in educating people and helping them to make a career transition.

=> Shashank Mishra :

~ Experienced Data Engineer with a demonstrated history of working in service and product companies. Solved data mysteries for different domains like Aviation, Pharmaceutical, FinTech, Telecom and Employee Services. Have designed scalable & optimized data pipelines to handle PetaBytes of data, with Batch & Real Time frequency. Got good exposure on different BigData frameworks (Hadoop, Spark, Hive, Sqoop, Flume, Flink, Kafka, Docker), Databases (MySQL, HBase, Cassandra, Redshift, Elastic Search), AWS Services (S3, Lambda, EMR, Glue, Cloudwatch, Redshift, SNS, SQS, Athena, Appflow), Dashboarding Tools (Grafana, Kibana, QuickSight, DataDog, Data Studio), Monitoring Tools (Airflow, Azkaban), Web Development (HTML, CSS, Scala Play, Django, Rest API, JavaScript, Ajax, JQuery), Good command over programming languages (Python, Java, Scala, Shell Scripting) and strong Data Structures & Algorithm fundamentals.

Curriculum details :-

=> Big Data Introduction: Introduction :

- ~ What is Big Data?
- ~ Evolution of Big Data
- ~ Why to learn Big Data technologies?
- ~ Examples of Big Data
- ~ Who is using Big Data?
- ~ Why is Data so important?
- ~ Characteristics of Big Data
- ~ Challenges of Big Data
- ~ Data scale
- ~ Manage, store and process Big Data
- ~ 5 Vs of Big Data
- ~ Sources of Data flood
- ~ Exploding data problem
- ~ OLTP and OLAP
- ~ Operational vs Analytical Big Data
- ~ Possible solutions: scaling up vs. scaling out
- ~ Challenges of scaling up and scaling out

=> Hadoop fundamentals :

- ~ What is Hadoop?
- ~ Hadoop in layman's term
- ~ History and timeline of Hadoop
- ~ Evolutionary features of Hadoop
- ~ Why hadoop in demand?
- ~ Components of Hadoop ecosystem
- ~ Hadoop architecture
- ~ How hadoop solve data explosion problem?
- ~ Differences between Hadoop 1.X and Hadoop 2.X and Hadoop 3.X
- ~ Hadoop 1.x 2.x 3.x architecture, components and working of those Components

=> HDFS :

- ~ Design of HDFS
- ~ HDFS architecture
- ~ HDFS features
- ~ Name node and data node
- ~ Secondary name node
- ~ Job tracker
- ~ Task tracker
- ~ Client nodes
- ~ Explain master-slaves
- ~ Pseudo-distributed
- ~ Fully-distributed
- ~ Data replication
- ~ How does a file read and write work?
- ~ Local file system and HDFS
- ~ Rack awareness
- ~ Arrangement of racks
- ~ Arrangement of machines and racks
- ~ Checkpointing in Hadoop
- ~ Benefits of replica placement and rack awareness
- ~ URL And URN
- ~ HDFS commands
- ~ HDFS web interface
- ~ Fault tolerance
- ~ Name node failure management
- ~ Anatomy of file read and write from HDFS
- ~ Important java classes to write data to HDFS
- ~ Inputsplit and data blocks difference
- ~ Why Is the block size 128 MB?
- ~ Recordreader
- ~ Inputformat
- ~ Default Inputformat:TextInputformat
- ~ Outputformat
- ~ What is partitioner?
- ~ Using partitioner
- ~ Map only job
- ~ Flow of operations in MapReduce
- ~ Serialization in MapReduce

=> HDFS Operations :

- ~ Start HDFS
- ~ Listing files in HDFS
- ~ Writing a file into HDFS
- ~ Reading data from HDFS
- ~ Shutting down HDFS
- ~ Listing contents of directory
- ~ Displaying and printing disk usage
- ~ Moving files & directories
- ~ Copying files and directories
- ~ Displaying file contents

=> YARN :

- ~ What is Yarn?
- ~ Why Yarn?
- ~ Classic MapReduce v/s Yarn
- ~ Yarn architecture

- ~ Resource Manager
- ~ Node manager
- ~ Application master
- ~ Node manager containers
- ~ Resource manager components
- ~ Advantages & disadvantages of Yarn
- ~ Yarn applications
- ~ Scheduling in Yarn
- ~ Fair Scheduler
- ~ Fault Tolerance
- ~ Schedulers in Yarn
- ~ FIFO scheduler
- ~ Capacity scheduler
- ~ Fair scheduler

=> Setting up Our Linux Space :

- ~ Downloading necessary tools
- ~ Installing Ubuntu in Windows
- ~ What is SSH?
- ~ Install SSH Clients
- ~ Setting up SSH in Ubuntu VM
- ~ How to do SSH to your Ubuntu VM?
- ~ Setting Up Passwordless SSH

=> AWS EC2 :

- ~ Launch a Basic EC2 Instance
- ~ Different Types of instances - Reserved, On-Demand, Spot, Dedicated
- ~ Different configurations of EC2 machines
- ~ Attaching detaching of EBS Volume in EC2
- ~ Practising few commands on EC2

=> AWS IAM :

- ~ The Mechanics behind IAM
- ~ Managing IAM Users
- ~ IAM Administration (Guide) (Listing, Deleting Users & Accounts)
- ~ Managing Permissions for IAM Users
- ~ Changing IAM User Permissions
- ~ Creating and Administering IAM Groups
- ~ Creating and Administering IAM Group Policies
- ~ Assigning Preset and Custom Group Policies

=> AWS S3 :

- ~ Buckets
- ~ Objects
- ~ Upload, Delete Files
- ~ Data Encryptions
- ~ Pricing & Data Limitation on S3
- ~ S3 Versioning
- ~ Version ID
- ~ Bucket policy
- ~ Notifications from S3
- ~ Work with S3 using AWS CLI
- ~ AWS Lambda
- ~ What is AWS Lambda and Why it is needed?
- ~ Features & Limitations of Lambda
- ~ Hello world program using Lambda
- ~ Auto trigger Lambda Function based on S3 file upload notification
- ~ Access other services from Lambda
- ~ AWS Secret Manager
- ~ Create and Maintain secrets
- ~ Accessing credentials from Secret Manager using Boto3

=> AWS EMR :

- ~ Setting up EMR Cluster
- ~ Install Spark, Hive, Hadoop
- ~ Resource types in EMR cluster
- ~ Data Processing on EMR Cluster
- ~ AWS Glue
- ~ Setting up cluster in Glue
- ~ Properties of Glue
- ~ Creating Catalogs in Glue
- ~ Read partitioned Data
- ~ Bulk and Incremental data processing from S3 in Glue
- ~ Data Processing in Glue
- ~ Glue jobs and Triggers

=> AWS SNS :

- ~ What is SNS?
- ~ How SNS works?
- ~ Creating SNS Topics and subscribing
- ~ Different types of subscribers
- ~ Sending notifications via SNS
- ~ AWS SQS
- ~ What is SQS?
- ~ Different types of SQS?
- ~ At-Least once and Exactly once delivery via SQS
- ~ Ingesting data to SQS
- ~ Inflight messages
- ~ Consume data from SQS
- ~ Dead Letter Queue

=> AWS DMS :

- ~ What is DMS?
- ~ Capturing CDC event in DMS where Database as a source
- ~ Capture CDC events and sending it to downstream systems
- ~ AWS Kinesis
- ~ Creating Kinesis Streams
- ~ Ingesting real time data in Kafka Streams
- ~ Consume real time data from Kafka Streams

=> AWS RDS :

- ~ MySQL Database using AWS RDS
- ~ Scalability & Limitations of AWS RDS
- ~ Creating tables and loading data in AWS RDS
- ~ Querying data from RDS
- ~ AWS Athena
- ~ What is serverless database services
- ~ Athena vs RDS
- ~ Table metadata in Athena for the data residing in S3
- ~ Creating table for S3 data
- ~ Querying S3 data using Athena

=> AWS Redshift :

- ~ What is Data warehousing services?
- ~ Architecture of Redshift
- ~ Resources types in Redshift Cluster
- ~ Creating tables in Redshift
- ~ Internal & External tables
- ~ Partitioning, Sort Keys, Column compression
- ~ Querying data in Redshift
- ~ Views & Materialized views in Redshift

=> AWS Dynamo :

- ~ Architecture of DynamoDB
- ~ Creating tables and Ingesting data into DynamoDB table
- ~ Querying data from DynamoDB
- ~ AWS Cloudwatch
- ~ Cron based triggers
- ~ Event pattern based triggers
- ~ Monitoring & Alerting using Cloudwatch
- ~ AWS QuickSight
- ~ Creating business dashboards using Quick sight

=> Introduction :

- ~ What is Hive?
- ~ Hive Vs Map Reduce
- ~ Hive Vs Relational databases
- ~ Installation and setup of Hive
- ~ Introduction to CouchDB
- ~ Why CouchDB?
- ~ History of CouchDB
- ~ Features of CouchDB
- ~ Advantages of CouchDB
- ~ Disadvantages of CouchDB
- ~ What is Neo4j?
- ~ Why Neo4j?
- ~ Features of Neo4j
- ~ Advantages of Neo4j
- ~ Neo4j Architecture
- ~ Applications of Neo4j
- ~ Data model of Neo4j
- ~ Building Blocks of Neo4j

=> Hive Architecture :

- ~ Hive architecture
- ~ Different modes of Hive
- ~ Hive Functions: Built-in & UDF
- ~ Datatypes in Hive
- ~ Operators in Hive
- ~ How to create and drop databases?
- ~ Hive create table: internal table, external table , alter, drop

=> DDL and DML commands in Hive :

- ~ Hive DDL
- ~ Create
- ~ Show
- ~ Describe
- ~ Use
- ~ Drop
- ~ Alter
- ~ Truncate
- ~ Hive DML
- ~ Load
- ~ Select
- ~ Insert
- ~ Delete
- ~ Update
- ~ Export
- ~ Import
- ~ Hive view and index
- ~ What is Hive metastore?

- ~ How to install and configure Hive metastore?
- ~ What is Hive data modeling?

=> Hive partitioning and bucketing :

- ~ Partitioning in Hive
- ~ Static and dynamic partitioning
- ~ Bucketing in Hive
- ~ Bucketing vs Partitioning
- ~ What is Hive query language(HQL)?

=> HQL language :

- ~ HiveQL- Where
- ~ HiveQL- Order By
- ~ HiveQL- Group By
- ~ HiveQL- Joins and types
- ~ HiveQL- SubQuery
- ~ Hive ETL: loading JSON, XML, text data
- ~ Working with arrays
- ~ Sort by and order by
- ~ Distribute by and cluster by
- ~ Bucket-map join
- ~ Sort-Merge-Bucket-Map join
- ~ Left semi join

=> Different File formats in Hive :

- ~ File formats in Hive
- ~ Text files
- ~ Input formats in Hive
- ~ Sequence files in Hive
- ~ RC file in Hive
- ~ Sequencefile
- ~ ORC files in Hive
- ~ Avro files
- ~ Parquet file
- ~ Inline index in ORC files
- ~ ORC file configurations in Hive
- ~ SerDe in Hive
- ~ Demo: CSVSerDe
- ~ JSONSerDe
- ~ RegexSerDe
- ~ Analytic and windowing in Hive
- ~ Demo: analytics.hql
- ~ Hcatalog in Hive
- ~ Demo: using_HCatalog
- ~ Accessing Hive with JDBC
- ~ Demo: HiveQueries.Java
- ~ HiveServer2 and beeline
- ~ Demo: beeline
- ~ Demo: ToUpper.Java and working_with_UDF
- ~ Optimizations in Hive
- ~ Demo: Optimizations

=> Introduction of HBase :

- ~ What is HBase?
- ~ HDFS and HBase
- ~ HBase vs RDBMS
- ~ HBase vs HIVE
- ~ HBase storage mechanism
- ~ Feature of HBase
- ~ Applications of HBase

=> HBase installation setup :

- ~ Apache HBase setup
- ~ Hardware recommendations
- ~ Software recommendations
- ~ Installation using cloudera manager
- ~ Basic static configuration

=> HBase architecture :

- ~ Architecture of HBase
- ~ Components of HBase architecture
- ~ Client library
- ~ Zookeeper
- ~ HMaster server
- ~ HBase regions servers

=> HBase commands :

- ~ General commands
- ~ status
- ~ table_help
- ~ version
- ~ whoami
- ~ Data definition commands
- ~ alter
- ~ alter_async
- ~ alter_status
- ~ create
- ~ drop
- ~ drop_all
- ~ enable
- ~ enable_all

- ~ exists
- ~ get_table
- ~ is_disabled
- ~ is_enabled
- ~ show_filters
- ~ Data manipulation commands
- ~ append
- ~ count
- ~ delete
- ~ deleteall
- ~ get_table
- ~ get_counter
- ~ put
- ~ truncate
- ~ truncate_preserve
- ~ Other HBase shell commands
- ~ Admin commands
- ~ Replication commands
- ~ Snapshot commands
- ~ Visibility labels commands
- ~ Security commands

=> CRUD operations using HBase shell :

- ~ What is HBase shell?
- ~ HBase shell usage
- ~ Starting HBase shell
- ~ Creating table
- ~ Inserting a row
- ~ Updating a row
- ~ Retrieving a row
- ~ Retrieving a range of rows
- ~ Deleting a row
- ~ Deleting a table
- ~ Retrieve rows within a time range
- ~ Filter by column value - SingleColumnValueFilter
- ~ Filter by Row id - RowFilter
- ~ Apply multiple conditions - Filterlist

=> Understanding the troubleshooting in HBase :

- ~ Understand the troubleshooting
- ~ Trouble shooting distributed clusters
- ~ Administration from the command line
- ~ How to use the HBase UI?
- ~ How to use the Metrics and the logs?

=> Basic Introduction :

- ~ Challenges with traditional RDBMS
- ~ What is Nosql database?
- ~ History behind the creation of Nosql databases
- ~ Features of Nosql database
- ~ Different types of Nosql databases
- ~ When Nosql should be used?
- ~ Advantages of Nosql
- ~ Disadvantages of Nosql
- ~ Why Nosql database?

=> Introduction and overview of cassandra :

- ~ What is Apache Cassandra?
- ~ History of Cassandra
- ~ Cassandra Database vs Relational Database
- ~ Apache Cassandra features
- ~ Cassandra use cases and applications
- ~ Advantages of Cassandra
- ~ Disadvantages of Cassandra

=> Setup, installtion and configuration :

- ~ Cassandra configuration with datastax
- ~ Understanding different ways to communicate with cassandra
- ~ Using cqlsh

=> Cassandra Architecture :

- ~ Cassandra architecture
- ~ Cassandra data model
- ~ Cassandra as a distributed database
- ~ Node
- ~ Data center
- ~ Cluster
- ~ Commit log
- ~ Mem-table
- ~ SSTable
- ~ Data replication
- ~ Write operation
- ~ Read operation
- ~ Data compaction

=> Cassandra Data Modeling :

- ~ Data modeling basics
- ~ Cassandra data modeling
- ~ Cassandra column types
- ~ Cassandra keyspace

=> Cassandra cluster and node :

- ~ *Configure and managing a cluster*
- ~ *Cluster and nodes*
- ~ *Adding nodes to cluster*
- ~ *Monitoring a cluster*
- ~ *Repairing a nodes*
- ~ *Removing a node*

=> Cassandra - Shell Commands :

- ~ *Help*
- ~ *Capture*
- ~ *Consistency*
- ~ *Copy*
- ~ *Describe tabel*
- ~ *Describe keyspaces*
- ~ *Expand*
- ~ *Exit*
- ~ *Show*
- ~ *Source*

=> Cassandra Query Language(CQL) :

- ~ *CQL Data Definition Commands*
- ~ *Cassandra CQL Data Types*
- ~ *Creating Database*
- ~ *Creating Keyspace*
- ~ *Use Keyspace*
- ~ *Alter Keyspace*
- ~ *Drop Keyspace*
- ~ *Create Table*
- ~ *Alter table*
- ~ *Drop table*
- ~ *Truncate*
- ~ *Create Index*
- ~ *Drop Index*
- ~ *CQL Data Manipulation Commands*
- ~ *Insert*
- ~ *Update*
- ~ *Delete*
- ~ *Batch*
- ~ *CQL Clauses*
- ~ *Select*
- ~ *Cassandra Where Clause*
- ~ *Cassandra Order by Clause*

=> Advanced CQL :

- ~ *CQL Collections*
- ~ *CQL User-Defined Types*
- ~ *Defining a Primary key*
- ~ *Defining a Partition key*
- ~ *Introduction to User-defined types(UDT)*
- ~ *How to Create a UDT?*
- ~ *UDT literals*
- ~ *How to alter a UDT?*
- ~ *How to drop a UDT?*

=> Cassandra CRUD Operation :

- ~ *Create data*
- ~ *Update data*
- ~ *Read data*
- ~ *Delete data*
- ~ *Maps*
- ~ *Sets*
- ~ *Lists*
- ~ *Key and indexing*

=> Introduction to MongoDB :

- ~ *Introduction*
- ~ *key charcristic of MongoDB*
- ~ *Understanding MongoDB ecosystem*
- ~ *Advantages & disadvantages of using MongoDB*

=> MongoDB installtion and setup :

- ~ *MongoDB installation in local*
- ~ *Setup MongoDB server*
- ~ *Setup MongoDB compass*
- ~ *Exploring the MongoDB compass*
- ~ *MondoDB local server and compass setup*
- ~ *MongoDB atlas setup*

=> Architecture :

- ~ *Architecture of MongoDB*
- ~ *Understanding databases, collections & documents*
- ~ *Creating databases & collections*
- ~ *Understanding JSON Data*
- ~ *Comparing JSON & BSON*
- ~ *Storage engines*
- ~ *Read path*
- ~ *Write path*
- ~ *Working set*
- ~ *Capped collection*

- ~ Oplog collection
- ~ TTL index
- ~ Gridfs

=> CRUD operations :

- ~ MongoDB data types
- ~ Finding, Inserting, Deleting & Updating elements
- ~ Querying the documents
- ~ Bulk insert operations
- ~ Updating multiple document
- ~ Limiting documents
- ~ Understanding insertOne vs insertMany()
- ~ Updateone() vs updateMany()
- ~ Understanding find() & fetchall()
- ~ Understanding "deleteOne()" & "deleteMany()"
- ~ Filtering documents

=> Schema design and data modeling :

- ~ Why do we use Schemas?
- ~ What is data modeling?
- ~ RDBMS and MongoDB data modeling difference
- ~ Embedding document
- ~ Reference document
- ~ Structuring documents
- ~ Understanding relations
- ~ One To One
- ~ One To Many
- ~ Many To Many

=> Database administration in MongoDB :

- ~ Database status
- ~ Troubleshooting issues
- ~ Current operations
- ~ Rotating log files
- ~ Users and roles
- ~ Copy and clone database
- ~ DB and collection stats
- ~ Explain plan
- ~ Profiling
- ~ Changing configuration files
- ~ Upgrading the database

=> MongoDB: backup and security :

- ~ Concept of backups
- ~ Mongoexport/mongoimport
- ~ Mongodump/mongorestore
- ~ Oplog backups
- ~ LVM backups
- ~ Backups using MMS/Ops manager
- ~ Purpose of security
- ~ Authentication and authorization
- ~ Role based access control

=> Working with python driver :

- ~ Splitting work between the Driver & the Shell
- ~ Preparing our project
- ~ Installing Visual Studio Code or Pycharm
- ~ Installing the Python
- ~ Connecting Python & the MongoDB cluster
- ~ Storing products in the database
- ~ Fetching data from the database
- ~ Getting a single product
- ~ Editing & deleting products
- ~ Implementing pagination
- ~ Adding an index
- ~ Adding an index to make the Email unique
- ~ Adding user sign-in

=> Replication in MongoDB :

- ~ Concept of replication
- ~ Replicaset member roles
- ~ Voting and electing primary
- ~ Role of oplog in replication
- ~ Read and write concern
- ~ Arbiter, Hidden and Delayed replica node
- ~ Priority settings
- ~ Replicaset nodes health check
- ~ Concept of resyncing the nodes
- ~ Rollbacks during failover
- ~ Keyfile authentication

=> MongoDB scalability :

- ~ Concept of scalability
- ~ Sharding concept
- ~ Shardkey and chunks
- ~ Choosing shardkey
- ~ Sharding components
- ~ Types of sharding
- ~ Balanced data distribution
- ~ Sharded and non-sharded collection
- ~ Sharded replicaset

- ~ Tag aware sharding

=> MongoDB Monitoring :

- ~ MMS manager
- ~ Ops manager
- ~ MongoDB utility commands
- ~ MongoDB developer tools
- ~ MongoDB client drivers

=> CouchDB Architecture :

- ~ CouchDB engine
- ~ HTTP request
- ~ Document
- ~ Replica database

=> Graph Database in Neo4j :

- ~ What is Graph Database?
- ~ Why Graph Database?
- ~ Graph DB Data Model
- ~ Graph DB vs RDBMS
- ~ The Property Graph Model

=> Setup :

- ~ Environment setup for Neo4j
- ~ Installation of Neo4j on Windows
- ~ Installation of Neo4j on Linux
- ~ Installation of Neo4j on Mac
- ~ Exploring Neo4j Bloom

=> Neo4j CQL :

- ~ Introduction to Neo4j CQL
- ~ Neo4j CQL clauses
- ~ Neo4j CQL Functions
- ~ Neo4j CQL Data Types
- ~ Neo4j CQL operators
- ~ Neo4j CQL Boolean operators
- ~ Neo4j CQL Comparison operators
- ~ Node Creation in Neo4j CQL
- ~ Relationship creation in Neo4j CQL

=> Neo4j CQL Operators :

- ~ Neo4j CQL Operators
- ~ Comparison Operators
- ~ Boolean Operators
- ~ String Operators
- ~ List Operators
- ~ Regular Expression
- ~ String matching

=> Neo4j clauses :

- ~ Match Clause
- ~ Optional Match Clause
- ~ Where Clause
- ~ Count Function
- ~ Return Clause
- ~ Order by Clause
- ~ Limit Clause
- ~ Skip Clause

=> Neo4j CQL Clauses :

- ~ Neo4j CQL General clauses
- ~ Neo4j CQL Write clauses
- ~ Neo4j CQL Readclauses

=> Introduction to Kafka :

- ~ Introduction to Apache Kafka
- ~ History of Apache Kafka
- ~ Why Apache Kafka?
- ~ What is messaging system?
- ~ Kafka message flow
- ~ Committed vs uncommitted messages
- ~ Kafka operations
- ~ Kafka communication
- ~ Advantages of Kafka
- ~ Kafka use-cases

=> Architecture of kafka :

- ~ Kafka architecture

=> Installation of kafka :

- ~ Installation of Kafka in local system
- ~ kafka setup on cloud
- ~ Kafka - Confluent
- ~ Kafka - Confluent platform

=> Kafka CLI :

- ~ Introduction to Kafka CLI
- ~ Creating Kafka topic
- ~ Listing topics in Kafka CLI
- ~ Deleting topics in Kafka CLI
- ~ Getting details of Kafka topic
- ~ Producing data to Kafka topic

- ~ Consuming data to Kafka topic
- ~ Purging a Kafka topic

=> Zookeeper in Kafka :

- ~ Why Zookeeper is used in Kafka?
- ~ Role of Zookeeper in Kafka

=> Kafka APIs :

- ~ Introduction to Kafka API
- ~ Different types of Kafka API
- ~ Producer API
- ~ Consumer API
- ~ Streams API
- ~ Connector API
- ~ Kafka integration with Spark

=> Introduction to NiFi :

- ~ What is Apache NiFi?
- ~ Architecture of Apache NiFi
- ~ Characteristics of Apache NiFi
- ~ Advantages of Apache NiFi

=> Installation of Apache NiFi :

- ~ Environment Setup
- ~ Setting up Windows Developer Environment
- ~ Setting up Linux Developer Environment
- ~ Setting up Mac Developer Environment

=> Apache NiFi Repository :

- ~ Flowfile Repository
- ~ Content Repository
- ~ Provenance Repository

=> Apache NiFi User Interface :

- ~ Introduction to Apache NiFi User Interface
- ~ NiFi Canvas
- ~ NiFi Processors
- ~ Process Groups and Templates
- ~ Apache NiFi components

=> Apache NiFi Processors :

- ~ Introduction to Apache NiFi Processors
- ~ GenerateFlowFile
- ~ LogAttribute
- ~ Functionality of NiFi Processors

=> Data Flows and Content :

- ~ NiFi Properties and Settings
- ~ Data Flow Monitoring in NiFi

=> Processing files in Apache NiFi :

- ~ Processing of CSV file
- ~ Processing of JSON file
- ~ Processing of Text file

=> Getting started with Spark :

- ~ What is Spark and what it is purpose?
- ~ Why Spark is faster than Hadoop?
- ~ What is in-memory computation?
- ~ Features of Spark
- ~ Explain unified architecture of Spark
- ~ Components of the Spark unified architecture
- ~ Downloading and installing Spark standalone
- ~ Scala and Python overview, launching and using Sparks Scala and Python shell
- ~ Spark execution context
- ~ Driver
- ~ Executor
- ~ Master
- ~ Worker

=> The Resilient Distributed Datasets (RDD) :

- ~ Overview of RDD's
- ~ Features of RDD
- ~ RDD operations
- ~ RDD and pair RDDs and RDD performance
- ~ Flat maps and filters
- ~ Data loading in RDD
- ~ RDD deep dive
- ~ Partitions
- ~ Dependencies
- ~ Transformation in RDD
- ~ Action in RDD
- ~ Map
- ~ Filter
- ~ Filter map
- ~ Group by
- ~ Group by key
- ~ Reduce by key
- ~ Map partitions
- ~ Union
- ~ Join
- ~ Distinct

- ~ Coalesce
- ~ Key by
- ~ Partition by
- ~ Zip
- ~ Collect
- ~ Reduce by key
- ~ Aggregate
- ~ RDD Lineage
- ~ DAG for RDD
- ~ Limitations of Spark RDD
- ~ RDD persistence
- ~ Shared variables and broadcast variables
- ~ Accumulators

=> Spark SQL, DataFrames and Datasets :

- ~ Introducing Spark SQL
- ~ Introducing datasets and DataFrame
- ~ Data sources
- ~ Distributed SQL engine
- ~ Creating DataFrame
- ~ DataFrame operations
- ~ DataFrame from csv
- ~ DataFrame from db tables
- ~ DataFrame from hive NoSQL tabel
- ~ DataFrame from json
- ~ DataFrame from RDD
- ~ Different operations on DataFrame
- ~ Filter
- ~ Join
- ~ Group
- ~ Aggregation
- ~ Having
- ~ Where
- ~ User define function(UDF)
- ~ Grouping aggregation
- ~ Multiple grouping
- ~ More aggregation
- ~ Hash aggregation
- ~ Spark SQL vs RDD
- ~ Executing SQL commands and SQL-style functions on a DataFrame
- ~ Using DataFrames instead of RDD's
- ~ Different operations with dataframes with DataFrames
- ~ Word Count with DataFrames
- ~ DataFrames vs RDDs
- ~ Operations on DFs
- ~ Parquet files with Spark Sql Read, Write, Partitioning, Merging schema
- ~ ORC files
- ~ JSON files

=> Spark streaming :

- ~ Basic concepts of Spark Streaming
- ~ Linking
- ~ Initializing Streaming Context
- ~ Discretized Streams (DStreams)
- ~ Input DStreams and Receivers
- ~ Transformations on DStreams
- ~ Output operations on DStreams
- ~ DataFrame and SQL operations
- ~ MLlib operations
- ~ Caching / Persistence
- ~ Checkpointing
- ~ Accumulators, Broadcast Variables, and Checkpoints
- ~ Deploying applications
- ~ Performance tuning
- ~ Reducing the batch processing times
- ~ Setting the right batch interval
- ~ Memory tuning
- ~ Sliding window operations
- ~ Overview Spark Streaming and Structure Streaming and kafka streaming with kafka
- ~ Developing Spark Streaming applications Integration with Hbase
- ~ Kafka Twitter data setup
- ~ Writing Producer in Python
- ~ Writing Consumer in Python
- ~ Kafka Integration with Spark Streaming
- ~ Fault-tolerance semantics
- ~ Spark Cassandra

=> Spark Structure streaming :

- ~ Handling Event-time and Late Data
- ~ API using Datasets and DataFrames
- ~ Creating streaming DataFrames and streaming Datasets
- ~ Input Sources
- ~ Schema inference and partition of streaming DataFrames/Datasets
- ~ Operations on streaming DataFrames/Datasets
- ~ Basic Operations - Selection, Projection, Aggregation
- ~ Window Operations on Event Time
- ~ Handling Late Data and Watermarking
- ~ Types of time windows
- ~ Join Operations

- ~ *Stream-static Joins*
- ~ *Stream-stream Joins*
- ~ *Inner Joins with optional Watermarking*
- ~ *Outer Joins with Watermarking*
- ~ *Semi Joins with Watermarking*
- ~ *Support matrix for joins in streaming queries*
- ~ *Streaming Deduplication*
- ~ *Policy for handling multiple watermarks*
- ~ *Arbitrary Stateful Operations*
- ~ *Unsupported Operations*
- ~ *Limitation of global watermark*
- ~ *State Store*
- ~ *HDFS state store provider*
- ~ *RocksDB state store implementation*
- ~ *State Store and task locality*
- ~ *Starting Streaming Queries*

=> **Launching on a clusters :**

- ~ *Spark standalone*
- ~ *Running Spark on Mesos*
- ~ *Running Spark on YARN*
- ~ *Running Spark on Kubernetes*
- ~ *The Spark Standalone Web UI*

=> **PySpark Installtion :**

- ~ *Installtion using PyPi*
- ~ *Pyspark setup in local*
- ~ *Pyspark setup with anaconda*
- ~ *Pyspark setup with pycharm*

=> **PySpark DataFrame :**

- ~ *DataFrame creation*
- ~ *Viewing data*
- ~ *Accessing data*
- ~ *Applying a function*
- ~ *Grouping data*
- ~ *Object creation*
- ~ *Missing data*
- ~ *Grouping*
- ~ *Plotting*

=> **Spark Mlib :**

- ~ *Overview of Mlib*
- ~ *What is Machine Learning?*
- ~ *Supervised learning*
- ~ *Unsupervised learning*
- ~ *Basic statistics*
- ~ *Classification algorithms*
- ~ *Regression algorithms*
- ~ *Clustering algorithms*
- ~ *Collaborative filtering*
- ~ *Frequent pattern mining*
- ~ *Featurization*
- ~ *Pipelines*
- ~ *Persistence*
- ~ *Spark ml for ml*
- ~ *Collect tranning data*
- ~ *Different proccessing technique*
- ~ *Supervised learning*
- ~ *Linear regression*
- ~ *Logistic regression*
- ~ *Decision tree navie bayes*
- ~ *Recommender system*
- ~ *End to end case study with real time dataset*

=> **GraphX :**

- ~ *Overview*
- ~ *Graph operations*
- ~ *Graph builders*
- ~ *Graph algorithms*

=> **Spark configuration, monitoring and tuning :**

- ~ *Understand components of spark cluster*
- ~ *configure spark to modify the spark properties, environmental variables, or logging properties*
- ~ *Monitor Spark using the web UIs, metrics, and external instrumentation*

=> **Installation of Drill :**

- ~ *Installing Apache Drill for Mac and Linux*
- ~ *Running Drill In non-embedded and cluster mode*
- ~ *Overview of Drill web console*

=> **Connecting to Data sources :**

- ~ *Connecting to local file system*
- ~ *Understanding storage plugins and workspaces*
- ~ *Connecting to MySQL*
- ~ *Connecting to Mongo*
- ~ *Connecting to Kafka*
- ~ *Connecting to Hive*
- ~ *Connecting to HBase*
- ~ *Querying across data sources*

=> Introduction to Sqoop :

- ~ *Sqoop introduction*
- ~ *How Sqoop works?*
- ~ *Why we use Sqoop?*
- ~ *Features of Sqoop*

=> Sqoop Tools :

- ~ *Sqoop architecture and working*
- ~ *Using command aliases*
- ~ *Controlling the Hadoop installation*
- ~ *Using generic and specific arguments*
- ~ *Using options files to pass arguments*

=> Sqoop import :

- ~ *Purpose of Sqoop import*
- ~ *Connecting to a database server*
- ~ *Selecting the data to import*
- ~ *Free-form query imports*
- ~ *Controlling the import process*
- ~ *Controlling transaction isolation*
- ~ *Controlling type mapping*
- ~ *Incremental imports*
- ~ *File formats*
- ~ *Large objects*
- ~ *Importing data into Hive*
- ~ *Importing data into HBase*
- ~ *Importing data into Accumulo*
- ~ *Connecting to a Mainframe*

=> Sqoop export :

- ~ *Purpose of Sqoop export*
- ~ *Inserts vs Updates*
- ~ *Exports and Transactions*

=> Sqoop - Job :

- ~ *Create Job*
- ~ *Verify Job*
- ~ *Inspect Job*
- ~ *Execute Job*

=> Validation in Sqoop :

- ~ *Introduction to the validation*
- ~ *Purpose of Validation*
- ~ *Limitations of Validations*

=> Setup of Airflow :

- ~ *Components of Airflow*
- ~ *Installing Airflow on mac*
- ~ *Installing Airflow on linux*
- ~ *Installing Airflow on windows*
- ~ *Run Airflow locally*
- ~ *Introduction to the Airflow UI*
- ~ *What you need to know about the UI*
- ~ *Introduction to the Airflow CLI*

=> Core concepts of Airflow :

- ~ *What is DAG?*
- ~ *DAG skeleton*
- ~ *Default arguments*
- ~ *Instantiate a DAG*
- ~ *Jinja templating with Airflow*
- ~ *What are tasks?*
- ~ *What are operators?*
- ~ *How to setup dependencies?*
- ~ *What are hooks*
- ~ *What are executors*

=> Loading data to Data Warehouse :

- ~ *Set up*
- ~ *Connections*
- ~ *Load data from storage*
- ~ *Run SQL query*
- ~ *Use a hook to list storage objects.*
- ~ *Cross-Task communication (XComs)*
- ~ *Variables*

=> Advanced concepts in Airflow :

- ~ *Adios repetitive patterns*
- ~ *Minimising DAG's with SubDAG's*
- ~ *Adios SubDAG's, welcome taskgroups!*
- ~ *Sharing data between tasks with xcoms*
- ~ *Choosing a specific path in your DAG*
- ~ *How Tasks get triggered?*

=> Creating Plugins with Elasticsearch and PostgreSQL :

- ~ *Installation of Elasticsearch*
- ~ *How the plugin system works?*
- ~ *Creating a Hook interacting with Elasticsearch*

=> Testing Airflow DAGS's :

- ~ *Load test DAG's*

- ~ Unit test DAG's and operators
- ~ Unit test custom operators

=> Docker Image for Apache Airflow :

- ~ Introduction to Docker
- ~ Why custom image?
- ~ How to build your own image?
- ~ Extending vs. customizing the image
- ~ Executors
- ~ Configure celery executors
- ~ Running Airflow on docker with celery executor
- ~ Configure local executors
- ~ Running Airflow on docker with local executor
- ~ Service level agreement
- ~ Security: Authentication, Roles, Encryption

=> Monitoring Airflow :

- ~ Airflow monitoring with StatsD
- ~ Airflow monitoring with Prometheus
- ~ Airflow monitoring with Graphana
- ~ Error tracking with Sentry

=> Introduction to Zookeeper :

- ~ Introduction of Apache Zookeeper
- ~ Why we need Zookeeper?
- ~ What is Distributed system?

=> Internal structure :

- ~ Zookeeper Background
- ~ Architecture Diagram
- ~ Important Components

=> Data models and Znodes :

- ~ Data model and Znode structure
- ~ What is Apache Zookeeper Znodes?
- ~ Sessions and watches

=> Installation of Zookeeper :

- ~ Installation of Apache zookeeper
- ~ Configuration of Apache zookeeper
- ~ Starting Apache zookeeper server
- ~ CLI operations

=> Monitoring in Zookeeper - Kafka :

- ~ Operating system
- ~ JMX monitoring

=> Installation and setup oozie :

- ~ Installation of oozie on your machine

=> Introduction to Ambari :

- ~ What is Apache Ambari?
- ~ Overview of Apache Ambari
- ~ History of Apache Ambari
- ~ Goals of Apache Ambari
- ~ Features of Apache Ambari
- ~ Benefits of Apache Ambari
- ~ Why should you learn Apache Ambari?
- ~ Apache Ambari architecture
- ~ Internal workflow of Ambari

=> Core applications of Ambari :

- ~ Server
- ~ Agent
- ~ Web UI
- ~ Database

=> Ambari usage :

- ~ Provisioning of Hadoop cluster
- ~ Monitoring of Hadoop cluster
- ~ Management of Hadoop cluster

=> How is Ambari is different from Zookeeper? :

- ~ Basic task
- ~ Nature
- ~ Status maintenance

=> Introduction to Cloud Databricks :

- ~ Introduction about cloud
- ~ Why cloud is important
- ~ Introduction to Databricks
- ~ Creating zure free account
- ~ Azure Databricks architecture overview

=> Databricks clusters :

- ~ Overview of clusters
- ~ Azure Databricks cluster types
- ~ Azure Databricks cluster configuration
- ~ Creating Azure Databricks cluster
- ~ Azure Databricks cluster pool

=> Mounting Data Lake container to databricks :

- ~ Overview

- ~ *Databricks file system (DBFS)*
- ~ *Databricks mount overview*
- ~ *Creating Azure data lake storage*
- ~ *Creating Azure service principal*
- ~ *Mounting Azure data lake Storage*
- ~ *Secret scopes overview*
- ~ *Creating secret scope In key vault*
- ~ *Mounting Data lake using secrets*

=> Data ingestion - CSV files :

- ~ *Data ingestion overview*
- ~ *What is circuits file*
- ~ *Requirements*
- ~ *DataFrame reader*
- ~ *Select columns*
- ~ *DataFrame writer*

=> Data ingestion - JSON files :

- ~ *What is JSON File?*
- ~ *Write data*

=> Introduction to Atlas :

- ~ *What is Apache Atlas*
- ~ *Features of Apache Atlas*

=> Installation Atlas :

- ~ *Installation of Apache Atlas*

=> Atlas terminology :

- ~ *Relationships*
- ~ *Attributes*
- ~ *System specific typews*
- ~ *Data lineage*
- ~ *Classification*

=> Atlas UI :

- ~ *Basic search in Atlas UI*
- ~ *Advanced search in Atlas UI*
- ~ *What is a glossary term?*
- ~ *Use cases of glossary*

=> Introduction to Confluent :

- ~ *Overview of Confluent*
- ~ *Features of Confluent*

=> Getting started with Confluent :

- ~ *Free trail for Confluent cloud*
- ~ *Quick start for Apache Kafka using confluent cloud*
- ~ *Confluent cloud console basics*
- ~ *KSQLDB in confluent cloud*
- ~ *Manage schemas on confluent cloud*
- ~ *REST API quick start for confluent cloud developers*

=> Kafka Clusters :

- ~ *Features and limits by cluster type*
- ~ *Create a cluster with a console*
- ~ *Expand a dedicated cluster with console*
- ~ *Shrink a dedicated cluster with console*
- ~ *Cluster management API overview*
- ~ *Migrate topics on confluent cloud clusters*

=> Manage topics in cloud console :

- ~ *Overview*
- ~ *Create, edit and delete topics*
- ~ *Use the message browser*

=> Stream governance :

- ~ *Overview*
- ~ *Stream Lineage*
- ~ *Stream Catalog*
- ~ *Stream Quality*

=> Cluster linking :

- ~ *Overview*
- ~ *Quick tutorial*
- ~ *Share Data Across Clusters, Regions And Clouds*
- ~ *Mirror topics*
- ~ *Data migration*

=> Confluent platform with cloud :

- ~ *Confluent platform with cloud*
- ~ *Connecting Control Center To Confluent Cloud*
- ~ *Connect Clients to Confluent Cloud*
- ~ *Connecting Kafka to Confluent Cloud*
- ~ *Connecting Kafka Streams to Confluent Cloud*

=> Confluent cloud API :

- ~ *Confluent Cloud API*
- ~ *Metrics API*

=> Confluent CLI :

- ~ *Installing Confluent CLI*
- ~ *Configuring Confluent CLI*

~ *Manage Confluent*

=> Introduction to AWS :

- ~ *What is AWS?*
- ~ *AWS solutions for BigData?*
- ~ *What is Data ingestion?*

=> Cloud computing on AWS :

- ~ *What is cloud computing?*
- ~ *Cloud services by AWS*
- ~ *Cloud Computing Tools on AWS*
- ~ *Cloud Computing Tools Pricing*
- ~ *Introduction to AWS S3*
- ~ *Creating your First S3 bucket*
- ~ *Uploading an object to your Bucket*
- ~ *Download an object*
- ~ *Copy your object to a Folder*
- ~ *Delete your object and Bucket*

=> AWS Storage :

- ~ *Introduction to AWS storage*
- ~ *What is Simple storage Service (S3)?*
- ~ *How S3 works?*
- ~ *Use cases of S3*
- ~ *Storage Hierarchy in S3*
- ~ *Buckets in S3*
- ~ *S3 pricing*
- ~ *Creating and S3 bucket*
- ~ *Uploading objects to the S3 bucket*
- ~ *What is Amazon S3 Glacier?*
- ~ *Glacier Vaults*
- ~ *Glacier Archives*
- ~ *Accessing Amazon S3 Glacier*

=> AWS Databases :

- ~ *Enabling object versioning in the S3 bucket*
- ~ *Databases on AWS*
- ~ *Introduction to Amazon Relational Database Service(RDS)*
- ~ *Features of RDS*
- ~ *Engine types Configuration*
- ~ *RDS Pricing*
- ~ *Creating a SQL Server DB Instance*
- ~ *Introduction to Amazon Aurora*
- ~ *Benefits of Amazon Aurora*
- ~ *Create an Aurora DB cluster*
- ~ *Introduction to Amazon Dynamo DB*
- ~ *Components of DynamoDB*
- ~ *Creating a DynamoDB table.*
- ~ *Connecting to the DB Instance From Your Machine*
- ~ *DynamoDB Items and Indexes*
- ~ *Dynamo Backup and Restore*

=> Collection :

- ~ *Collection*
- ~ *Collection Section Introduction*
- ~ *Kinesis Data Streams Overview*
- ~ *Hot shard*
- ~ *Kinesis Producers*
- ~ *Kinesis Consumers*
- ~ *Kinesis Enhanced Fan Out*
- ~ *Kinesis Scaling*
- ~ *Kinesis - Handling Duplicate Records*
- ~ *Kinesis Security*
- ~ *Kinesis Data Firehose*
- ~ *CloudWatch Subscription Filters with Kinesis*
- ~ *SQS Overview*
- ~ *SQS Hands On*
- ~ *Kinesis Data Streams vs SQS*
- ~ *IoT Overview*
- ~ *IoT Components Deep Dive*
- ~ *Database Migration Service (DMS)*
- ~ *Direct Connect*
- ~ *AWS Snow Family Overview*
- ~ *AWS Snow Family Hands On*
- ~ *MSK: Managed Streaming for Apache Kafka*
- ~ *Kinesis vs MSK*

=> Storage :

- ~ *S3 Overview*
- ~ *S3 Hands On*
- ~ *S3 Security: Bucket Policy*
- ~ *S3 Security: Bucket Policy Hands On*
- ~ *S3 Website Overview*
- ~ *S3 Website Hands On*
- ~ *S3 Versioning Overview*
- ~ *S3 Versioning Hands On*
- ~ *S3 Server Access Logging*
- ~ *S3 Server Access Logging Hands On*
- ~ *S3 Replication Overview*
- ~ *S3 Replication Hands On*

- ~ S3 Storage Classes Overview
- ~ S3 Storage Classes Hands On
- ~ S3 Glacier Vault Lock & S3 Object Lock
- ~ S3 Encryption
- ~ Shared Responsibility Model for S3
- ~ DynamoDB Overview
- ~ DynamoDB RCU & WCU
- ~ DynamoDB Partitions
- ~ DynamoDB APIs
- ~ DynamoDB Indexes: LSI & GSI
- ~ DynamoDB DAX
- ~ DynamoDB Streams
- ~ DynamoDB TTL
- ~ DynamoDB Security
- ~ DynamoDB: Storing Large Objects

=> Processing :

- ~ Section Introduction: Processing
- ~ Lambda Overview
- ~ Lambda Hands On
- ~ [Exercise] AWS Lambda
- ~ Why Cloud & Big Data on Cloud
- ~ What is Virtual Machine
- ~ On-Premise vs Cloud Setup
- ~ Major Vendors of Hadoop Distribution
- ~ Hdfs vs S3
- ~ Important Instances in AWS
- ~ Spark Basics
- ~ Why spark is difficult
- ~ Overview of EMR
- ~ What is EMR
- ~ Tez vs mapreduce
- ~ Launching an emr cluster
- ~ connecting to your cluster
- ~ Create a tunnel for web ui
- ~ Use Hue to interact with EMR
- ~ Transient vs Long Running Cluster Running
- ~ Copy File From S3 to Local Zeppelin Notebook
- ~ How to Create a VM
- ~ S3 & EBS
- ~ Public ip Vs Private Ip
- ~ Aws Command Line Interface
- ~ AWS Glue
- ~ Introduction to Amazon Redshift
- ~ Redshift Master Slave Architecture
- ~ redshift demo
- ~ redshift spectrum
- ~ Redshift Distribution Styles
- ~ Redshift Fault Tolerance
- ~ Redshift Sort Keys

=> Analysis :

- ~ Section Introduction: Analysis
- ~ Intro to Kinesis Analytics
- ~ Kinesis Analytics Costs; RANDOM_CUT_FOREST
- ~ Intro to Opensearch (formerly Elasticsearch)
- ~ Amazon Opensearch Service
- ~ Opensearch Features
- ~ What is Athena
- ~ When do we require Athena What problem Athena Solve How Athena Works
- ~ Athena Pricing
- ~ Athena Practical Demonstration

=> Visualization :

- ~ The course overview
- ~ big data analytics and aws
- ~ How Quicksight is different than other BI Tools
- ~ BI solution based on quicksight
- ~ how to get started with quicksight
- ~ Performance Your first analysis
- ~ AWS Big data ecosystem
- ~ importing files to quicksight
- ~ importing databases to quicksight
- ~ importing data from saas services to quicksight
- ~ edit existing data sources in quicksight
- ~ Joining datasets
- ~ using functions
- ~ applying filters
- ~ understanding spice layer
- ~ Creating a Quicksight Analysis
- ~ Explore various charting options
- ~ Exploring various Map options
- ~ Exploring various table and other visual options
- ~ Mini project Overview
- ~ Mini Project Architecture
- ~ Data ingestion for mini project
- ~ Reports and dashboards

=> Introducing Google Cloud Platform :

- ~ Google platform fundamentals overview.

~ Google cloud platform Big Data products.

=> Compute and Storage Fundamentals :

~ CPUs on demand (compute engine).

~ A global filesystem (cloud storage).

~ CloudShell.

~ Set up an Ingest-Transform-Publish data processing pipeline.

=> Data Analytics on the Cloud :

~ Stepping-stones to the cloud.

~ Cloud SQL: your SQL database on the cloud.

~ Importing data into CloudSQL and running queries.

~ Spark on Dataproc.

~ Machine Learning recommendations with Spark on Dataproc.

=> Scaling Data Analysis :

~ Fast random access.

~ Datalab

~ BigQuery.

=> Data Processing Architectures :

~ Message-oriented architectures with Pub/Sub.

~ Real time streaming using Pub/Sub

~ Creating pipelines with Dataflow.

~ Reference architecture for real-time and batch data processing.

~ Google Online Transfer

~ Cloud Storage Transfer

~ Google Cloud BigTable

~ Google Cloud Dataflow

~ Google Cloud Dataproc

~ Google Cloud Pub/Sub

~ Google Cloud Composer

~ Google Cloud Data Fusion

~ Automating ETL jobs with composer and fusion

~ Google Cloud data catalog

~ Google Data studio

~ Architecture: Optimizing large-scale Ingestion

~ GCP Big Data Outro

=> Introduction to cloud :

~ Introduction to Cloud Computing

~ Cloud models

~ Different cloud providers

=> Regions and Availability Zones :

~ Understanding regions and availability zones in Azure

~ Creating microsoft Azure account

=> Resource Hierarchy :

~ Understanding resource hierchy

~ Demo on resource hierchy

~ Resource groups, subscription and managment groups

~ Active directory

=> Introduction to azure cloud computing :

~ Azure services overview

~ Managed and unmanaged service

~ Demo create Azure SQL Database service

=> Introduction to data engineer profile :

~ Introduction to data engineer Technologies

~ Data engineer role and responsibility

~ Introduction to data engineer technologies

=> Azure sql database :

~ Module Introduction

~ Introduction

~ Why choosing SQL Server in Azure

~ Azure laas vs Paas database offerings

~ SQL server paas deployment options

~ Introduction to Azure sql server in virtual machine

~ SQL Server in Azure virtual machine

~ SQL server in azure virtual machine

~ Introduction Azure single database

~ Demo Azure single database

~ Purchasing Models and service tier

~ Azure database vs azure datawarehouse

~ Introduction elastic data pool

~ Azure Elastic Database

~ Azure Elastic Database

~ Introduction managed instance Database

~ Azure managed instance Database

~ Difference between on premises and manged instance

~ Service tiers for managed instance

~ Mangement operations

~ Demo managed instance

=> Azure synapse :

~ Module introduction

~ Why warehouse in cloud?

~ Traditional vs modern warehouse architecture

~ What is synapse analytics service?

- ~ Demo create dedicated sql pool
- ~ Demo connect sql pool with ssms
- ~ Demo create Azure synapse analytics workspace
- ~ Demo explore synapse studio v2
- ~ Demo create dedicated sql pool and spark pool from inside synapse studio
- ~ Demo analyse data using dedicated sql pool
- ~ Analyse data using apache spark notebook
- ~ Demo analyse data using serverless sql
- ~ Demo data factory copy tool from synapse integrate tab
- ~ Demo monitor synapse analytics studio
- ~ Azure synapse a game changer
- ~ Azure synapse benefits

=> Azure databricks :

- ~ Spark Basics
- ~ Why spark is difficult?
- ~ Why databricks in cloud?
- ~ How to save databricks demo cost
- ~ Demo provision databricks, clusters and workbook
- ~ Demo mount data lake to databricks DBFS
- ~ Demo Explore, Analyze, Clean, Transform and load data in databricks
- ~ Azure databricks cluster
- ~ Azure databricks other important components
- ~ Databricks monitoring

=> Azure data factory :

- ~ What is Data Factory?
- ~ Data factory in azure ecosystem
- ~ Provision Azure data factory instance
- ~ Data factory components
- ~ Data factory pipeline and activities
- ~ Data factory linked service and datasets
- ~ Data factory integration runtime
- ~ Data factory triggers
- ~ Data factory copy data activity demo
- ~ Copy data activity using author demo
- ~ Secure input and output property
- ~ User properties
- ~ Data factory parameters
- ~ Data flow concept
- ~ Mapping data flow
- ~ Wrangling data flow
- ~ Monitoring
- ~ Metrics and diagnostic settings

=> Introduction to SQL :

- ~ Why SQL?
- ~ Application of SQL
- ~ Characteristics of SQL
- ~ Installation guide
- ~ Connection & set up
- ~ Create database
- ~ RENAME database
- ~ Drop database
- ~ SELECT database

=> Data type of SQL :

- ~ Binary datatypes
- ~ Approximate numeric datatype
- ~ Exact numeric datatype
- ~ Character string datatype
- ~ Date and time datatype

=> Introduction to SQL syntax :

- ~ SQL SELECT statement
- ~ SQL WHERE clause
- ~ SQL DISTINCT clause
- ~ SQL AND/OR clause
- ~ SQL IN clause
- ~ SQL LIKE clause
- ~ SQL BETWEEN clause
- ~ SQL ORDER BY clause
- ~ SQL GROUP BY clause
- ~ SQL COUNT clause
- ~ SQL HAVING clause
- ~ SQL CREATE TABLE statement
- ~ SQL DROP TABLE statement
- ~ SQL CREATE INDEX statement
- ~ SQL DROP INDEX statement
- ~ SQL DESC statement
- ~ SQL TRUNCATE TABLE statement
- ~ SQL ALTER TABLE statement
- ~ SQL ALTER TABLE statement(rename)
- ~ SQL INSERT INTO statement
- ~ SQL UPDATE statement
- ~ SQL DELETE statement
- ~ SQL CREATE DATABASE statement
- ~ SQL DROP DATABASE statement
- ~ SQL USE statement
- ~ SQL COMMIT statement

~ SQL ROLLBACK statement

=> Operators in SQL :

- ~ Arithmetic operators
- ~ Comparison operators
- ~ Logical operators
- ~ Operators used to negate conditions

=> SQL Query :

- ~ CREATE table
- ~ CREATE table with PRIMARY KEY
- ~ CREATE table with FOREIGN KEY
- ~ DELETE table
- ~ TRUNCATE table
- ~ TEMP table
- ~ RENAME table
- ~ DROP table
- ~ COPY table
- ~ ALTER table
- ~ INSERT query
- ~ UPDATE query
- ~ DELETE query

=> SELECT Query :

- ~ SELECT statement
- ~ SELECT UNIQUE
- ~ SELECT DISTINCT
- ~ SELECT COUNT
- ~ SELECT TOP
- ~ SELECT LAST
- ~ SELECT RANDOM
- ~ SELECT IN
- ~ SELECT RANDOM
- ~ SELECT MULTIPLE
- ~ SELECT DATE
- ~ SELECT SUM
- ~ SELECT NULL
- ~ SELECT group by

=> SQL Clause :

- ~ WHERE clause
- ~ AND clause
- ~ OR clause
- ~ WITH clause
- ~ AS clause
- ~ HAVING clause
- ~ Like clause
- ~ IS NULL clause
- ~ UNION clause
- ~ UNION All clause
- ~ Top clause

=> SQL Order By :

- ~ ORDER BY clause
- ~ ORDER BY ASC
- ~ ORDER BY DESC
- ~ ORDER BY

=> SQL Constraints :

- ~ NOT NULL constraint
- ~ DEFAULT constraint
- ~ UNIQUE constraint
- ~ PRIMARY key
- ~ FOREIGN key
- ~ CHECK constraint
- ~ INDEX
- ~ Introduction to views

=> Functions(Aggregate) :

- ~ Conditional aggregation
- ~ List concatenation
- ~ SUM
- ~ AVG()
- ~ Count
- ~ Min
- ~ Max

=> SQL Joins :

- ~ INNER JOIN
- ~ LEFT JOIN
- ~ RIGHT JOIN
- ~ FULL JOIN
- ~ SELF JOIN
- ~ CARTESIAN JOIN

=> Views in SQL :

- ~ Creating view
- ~ Creating view from single table
- ~ Creating view from multiple tables
- ~ Delete view

=> Window Functions :

- ~ Setting up a flag if other rows have a common property
- ~ Finding "Out-of-Sequence" records using the LAG() function
- ~ Getting a running total
- ~ Adding the total rows selected to every row
- ~ Getting the N most recent rows over multiple grouping

=> Subqueries :

- ~ Subquery in FROM clause
- ~ Subquery in SELECT clause
- ~ Subquery in WHERE clause
- ~ Correlated subqueries
- ~ Filter query results using query on different table
- ~ Subqueries in FROM clause
- ~ Subqueries in WHERE clause

=> Stored Procedures :

- ~ Create and call a stored procedure
- ~ In and out parameters
- ~ If, Elseif and Else
- ~ Case
- ~ While
- ~ Repeat
- ~ Cursor
- ~ Loop
- ~ Error handling
- ~ User defined errors
- ~ Transactions
- ~ Stored functions

=> Triggers :

- ~ CREATE TRIGGER
- ~ Use trigger to manage a "Recycle Bin" for deleted items

=> AWS Lambda :

- ~ What is AWS Lambda and Why it is needed?
- ~ Features & Limitations of Lambda
- ~ Hello world program using Lambda
- ~ Auto trigger Lambda Function based on S3 file upload notification
- ~ Access other services from Lambda

=> AWS Secret Manager :

- ~ Create and Maintain secrets
- ~ Accessing credentials from Secret Manager using Boto3

=> AWS Glue :

- ~ Setting up cluster in Glue
- ~ Properties of Glue
- ~ Creating Catalogs in Glue
- ~ Read partitioned Data
- ~ Bulk and Incremental data processing from S3 in Glue
- ~ Data Processing in Glue
- ~ Glue jobs and Triggers

=> AWS SQS :

- ~ What is SQS?
- ~ Different types of SQS?
- ~ At-Least once and Exactly once delivery via SQS
- ~ Ingesting data to SQS
- ~ Inflight messages
- ~ Consume data from SQS
- ~ Dead Letter Queue

=> AWS Kinesis :

- ~ Ingesting real time data in Kafka Streams
- ~ Consume real time data from Kafka Streams

=> AWS Cloudwatch :

- ~ Cron based triggers
- ~ Event pattern based triggers
- ~ Monitoring & Alerting using Cloudwatch

=> AWS QuickSight :

- ~ Creating business dashboards using Quick sight

Amazon Lex

Topic Name : DATA SCIENCE

Sub-topic Name : CHATBOT

Course link : <https://ineuron.ai/course/Amazon-Lex>

Course Description :-

This course is designed for creating intelligent chatbot interfaces with AWS (Application Web Services). We'll make our hands dirty and take a challenge in which we have to develop a sample chatbot together; after that, we'll learn how to make our bot available on our Facebook Page, Slack, WhatsApp, Telegram.

Course Features :-

- => Basics of Chatbot
- => Integration of your bot in Facebook, Telegram, slack
- => Challenges
- => Source code
- => Downloadable resources
- => Quizzes
- => Completion Certificate

What you will learn :-

- => Complete architecture of Amazon lex
- => How to create one hybrid conversational interfaces (i.e. chatbots) that work in any of the platforms like in Facebook Messenger, Skype, Slack, Telegram, and many more
- => Create business ready chatbots

Requirements :-

- => AWS Account
- => No prior experience with Chatbot
- => Slack, Facebook, Telegram accounts
- => A system with Internet Connection.
- => Your Dedication

Instructors :-

=> Khushali Shah :

~ A data scientist having rich experience working with MNCs and start-ups in the field of data science and machine learning. She has expertise in Chatbot development for various domains & been developing professionally for 6+ years with diverse job history. She also had positions in software module development, web app development, functional designs, requirement gathering, client interaction, and server setup/admin & can help everywhere in the stack; she loves wearing multiple hats to an extent. She also believes in enhancing her skills by training and learning new things day by day.

Curriculum details :-

- => Course Introduction :
 - ~ Introduction Preview
 - ~ What is Chatbot?
 - ~ Why Chatbot?
 - ~ Get your AWS account ready
- => Amazon Lex Fundamentals :
 - ~ What is Lex Preview
 - ~ Lex supported languages
 - ~ Programming modal
 - ~ Intent/Slot
 - ~ Lex model building API
 - ~ Runtime API Operations
 - ~ Managing messages
 - ~ Confidence score
 - ~ Conversation history log
 - ~ Built-in intent
 - ~ Built-in slot
 - ~ Custom slot
 - ~ Sentiment analysis
 - ~ Lambda function schema
 - ~ Integration
- => Custom Building Chatbot :
 - ~ Chatbot Problem Statement
 - ~ Create Intents
 - ~ JokeIntent
 - ~ Lambda

- ~ *Facebook integration*
- ~ *Slack integration*
- ~ *Slack app*

=> Course Summary :

- ~ *Summary*

FastAPI

Topic Name : WEB DEVELOPEMENT

Sub-topic Name : FAST API

Course link : <https://ineuron.ai/course/FastAPI>

Course Description :-

FastAPI is a modern, high-performance, web framework for building APIs with Python. This course is specifically developed for beginners! This implies that irrespective of your background, you will be able to master one of the most popular frameworks on the market. All you need is a basic understanding of Python.

Course Features :-

- => Completion Certificate
- => Quiz in every module
- => Real-time Implementations

What you will learn :-

- => Introduction to API
- => Getting Started with FastAPI
- => Features of FastAPI
- => Python Type Hints

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Instructors :-

=> Monal Kumar :

~ Monal Kumar is a data scientist and instructor working at iNeuron having 2+ years of total experience in both service and product-based organisations. He is specialised in Deep Learning, Computer vision and Image processing. Previously, he held positions as a support configurator at Wipro Technologies and as a Deep Learning researcher at Harptec Research. Offering the finest possible services to his clients. In addition to his primary job function, he is recognised for his creativity and ideas that change the nature of the existing problem.

Curriculum details :-

=> Course Introduction :

- ~ Welcome to FastAPI course
- ~ Course pre-requisites
- ~ Who is this course for
- ~ What you will get from this course
- ~ What is API
- ~ Why API is used
- ~ Different ways to create API
- ~ Advantage of API
- ~ How to get access to course materials
- ~ What career path you can follow after completion

=> Getting Started with FastAPI :

- ~ What is FastAPI
- ~ Installing Python
- ~ Installing FastAPI
- ~ Your first 'hello world' program using default GET
- ~ GET and PUT
- ~ What is JSON format
- ~ JSON response
- ~ Practical GET and PUT
- ~ Interactive API docs
- ~ Alternative API docs
- ~ What is uvicorn
- ~ Features of using FastAPI

=> Python Hint Types :

- ~ Python Type hints introduction
- ~ Why they are important
- ~ Using hint types in function
- ~ Specifying function return type

=> FastAPI :

- ~ import FastAPI
- ~ Creating FastAPI Instance
- ~ URL
- ~ HTTP methods
- ~ What is decorator

- ~ *Defining a decorator*
- ~ *Defining path operation function*
- ~ *Path parameter*
- ~ *Path parameter with types*
- ~ *Working with python enumeration*
- ~ *Query parameters*
- ~ *Multiple path and query parameters*
- ~ *Request Body*
- ~ *Query Parameters and String Validations*
- ~ *Form data*
- ~ *Request files*
- ~ *Practical Demo Request forms and files*
- ~ *Handling errors*

=> Summary :

- ~ *Course Outro*

Hive Project to build a data warehouse for e-Commerce

Topic Name : BIG DATA

Sub-topic Name : BIG DATA PROJECTS

Course link : <https://ineuron.ai/course/Hive-Project-to-build-a-data-warehouse-for-e-Commerce>

Course Description :-

In this project, we will learn how to build a complete end-to-end data warehouse for an e-commerce application to perform Apache Hive

analytics on various datasets using big data tools like Apache Sqoop, Apache Spark, and HDFS. In order to achieve this task, we utilize the various services provided by AWS. For this hive project, we will delve further into some of the analytical features of the Hive.

Course Features :-

- => Do Everything In Industry Grade Lab
- => Learn As Per Your Timeline
- => Hands-On Industry Real-Time Projects.
- => Self Paced Learning
- => Dashboard Access

What you will learn :-

- => Real Time Projects
- => Hive Project to build a data warehouse for e-Commerce
- => Understanding various services provided by AWS
- => Creating an AWS EC2 instance
- => Moving the data from MySQL to HDFS
- => Data ingestions using Sqoop, Spark and Hive
- => Understanding different analytical functions in Hive

Requirements :-

- => System with minimum i3 processor or better
- => At least 4 GB of RAM
- => Working internet connection
- => Dedication to learn

Instructors :-

=> Priya Bhatia :

~ Expertise in data structure competitive programing and solving an analytical problems and implementing data structure algorithm in multiple programming language. I have done my M.Tech in Artificial Intelligence at IIT Hyderabad and have an experience of implementation in multiple projects.

Curriculum details :-

- => Welcome to the Course :
 - ~ Course Overview
 - ~ Dashboard Introduction
- => Project :- Hive Project to build a data warehouse for e-Commerce :
 - ~ Introduction of Instructor
 - ~ Project Overview
 - ~ End Notes
 - ~ Problem Description
 - ~ Understand the application scope
 - ~ Tour to an existing solution
 - ~ End Notes
 - ~ Big Data
 - ~ Data warehouse
 - ~ MySQL
 - ~ Apache Hive
 - ~ Apache Sqoop
 - ~ Apache Spark
 - ~ End Notes
 - ~ Solution Description
 - ~ Tour of the architecture diagram
 - ~ Cost Involved
 - ~ End Notes
 - ~ Launch of AWS EC2 Instance
 - ~ Docker Image creation
 - ~ Environment Setup Steps in detail
 - ~ MySQL table creation
 - ~ Data ingestion: MySQL to HDFS via Apache Sqoop

- ~ *HDFS to Hive Data transfer and integration into Apache Spark*
- ~ *Extraction of customer demographics info*
- ~ *Parquet file storage and its usage*
- ~ *Transfer of parquet file from Spark to Hive*
- ~ *Perform Hive analytics on sales and customer data*
- ~ *Code Files and End Notes*
- ~ *Conclude the project*
- ~ *Assignments & External Resources*

Deep Learning for Kids

Topic Name : K12

Sub-topic Name : CLASS10

Course link : <https://ineuron.ai/course/Deep-Learning-for-Kids>

Course Description :-

Learners will master the fundamentals of deep learning as well as how to tackle a challenging real-world problem that is difficult to handle with standard programming in this course. This course will teach you the fundamentals of AI, allowing you to create incredible AI applications. Students will receive hands-on practical experience in designing AI-based projects after successfully completing the course. Learners might begin applying for freelancing employment in order to make a fortune.

Course Features :-

- => Online Instructor-led learning
- => Practical Implementation
- => Integrate academic knowledge with tech
- => Real-time project
- => Live class recording
- => Doubt clearing
- => Assignment in all the module
- => Quiz in every module
- => Career Counselling
- => Completion certificate

What you will learn :-

- => Introduction Artificial Intelligence
- => Introduction to Deep Learning
- => Supervised learning
- => Unsupervised learning
- => Python basics
- => NumPy basics
- => Pandas basics
- => TensorFlow
- => Kera's
- => Artificial neural network
- => Convolution neural network
- => Projects

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Curriculum details :-

- => Course Introduction :
 - ~ Welcome to machine learning course
 - ~ What you will learn from this course
 - ~ Course pre-requisites
 - ~ What is deep learning?
 - ~ Who is this course for?
 - ~ What you will get from this course?
 - ~ How to get access to course materials?
 - ~ What career path you can follow after completion of this course?
- => Introduction to AI :
 - ~ What is Artificial intelligence?
 - ~ History of AI
 - ~ Applications of AI
 - ~ Advantage of AI
 - ~ Practical use of AI
- => Introduction to learning :
 - ~ What do you mean by learning?

- ~ Why deep learning?
- ~ How babies learn: An Analogy
- ~ Different types of learning
- ~ What is Supervised learning?
- ~ Supervised learning example: Importance of Teacher feedback
- ~ What is Unsupervised learning?
- ~ Unsupervised learning example: Categorizing students based on hobbies (Annual Function)
- ~ What is Reinforcement learning?
- ~ How a self-driving car works: An Analogy
- ~ Discussion: Sofia robot
- ~ Uses of Deep learning

=> Assignment1 :

- ~ Give 3 examples of AI used in the education sector.

=> Preparing your system :

- ~ Why python?
- ~ Colab overview

=> Working with important libraries :

- ~ Python basics
- ~ Numpy basics
- ~ Pandas basics
- ~ Tensorflow basics
- ~ Keras basics

=> Assignment2 :

- ~ Create a function to add 2 numbers
- ~ Create a function that will take name and address from user and print the output
- ~ Print multiplication table of 1 to 10 using for loop
- ~ Print multiplication table of 1 to 10 using for loop
- ~ Using TensorFlow add two numbers and print the output
- ~ Using TensorFlow add two multiply to matrices and print the output

=> Neural network basics :

- ~ What is neuron?
- ~ Neural network vs Human brain network
- ~ What is perceptron?
- ~ What is ANN?
- ~ Practical: Perceptron
- ~ Tensorboard overview
- ~ Logging the activity of training using Tensorboard
- ~ Analysis: How to classify orange and apple with features
- ~ Practical: Predicting the price of premium phones for the year 2023 using Neural network
- ~ Explain Logistic Regression
- ~ Practical: Classifying male and female based on height and weight of a person
- ~ What do you mean by Activation function?

=> Assignment3 :

- ~ Create a neural network and predict the price of mobile network recharge for next 1 year

=> Convolution neural networks :

- ~ Introduction
- ~ What are images?
- ~ Image data vs numerical data
- ~ Practical: Deep neural network
- ~ Using Netron to visualize neural network
- ~ What is CNN?
- ~ Why use CNN instead of N-layer neural network?
- ~ Visualizing different layers of CNN using web app: <https://blog.terencebroad.com/archive/convnetvis/vis.html>
- ~ Practical: Basic CNN using keras
- ~ Practical: Create a CNN and identify day, evening and night
- ~ Discussion: Use cases of CNN (Detection, tracking)

=> Assignment4 :

- ~ Create a CNN to classify whether the room is empty or not

=> Projects :

- ~ Classifying apple vs orange
- ~ Fruit classification using CNN

=> Summary :

- ~ Course Outro
- ~ Future Scope of Deep learning

Full Stack Data Science Nov'21 Batch

Topic Name : DATA SCIENCE

Sub-topic Name : FULL STACK DATA SCIENCE

Course link : <https://ineuron.ai/course/Full-Stack-Data-Science-Nov'21-Batch>

Course Description :-

This is a data science full stack live mentor led certification program along with full time one-year internship provided by iNeuron intelligence private limited, where you will learn all the stack required to work in data science, data analytics and big data industry including ML ops and cloud infrastructure and real time industry project and product development along with iNeuron product development team and you will contribute on various level with iNeuron .

Course Features :-

- => Full stack Data Science masters certification
- => Job guarantee otherwise refund
- => One year of internship Anytime
- => Online Instructor-led learning: Live teaching by instructors
- => 56 + hands-on industry real-time projects.
- => 500 hours live interactive classes.
- => Every week doubt clearing session after the live classes.
- => Lifetime Dashboard access
- => Doubt clearing one to one
- => Doubt clearing through mail and skype support team
- => Assignment in all the module
- => Quiz in every module
- => A live project with real-time implementation
- => Resume building Anytime
- => Career guidance Anytime
- => Interview Preparation Anytime
- => Regular assessment
- => Job Fair and Internal Hiring
- => Mock Interview Anytime

What you will learn :-

- => Python
- => Stats
- => Machine learning
- => Deep learning
- => Computer vision
- => Natural language processing
- => Data analytics
- => Big data
- => ML ops
- => Cloud
- => Data structure and algorithm
- => Architecture
- => Domain wise project
- => Databases
- => Negotiations skills
- => Mock interview
- => Interview preparation
- => Resume building after every module

Requirements :-

=> Dedication

=> Computer with i3 and above configuration

Instructors :-

=> krish naik :

~ Having 10+ years of experience in Data Science and Analytics with product architecture design and delivery. Worked in various product and service based Company. Having an experience of 5+ years in educating people and helping them to make a career transition.

=> Sudhanshu Kumar :

~ Having 8+ years of experience in Big data, Data Science and Analytics with product architecture design and delivery. Worked in various product and service based Company. Having an experience of 5+ years in educating people and helping them to make a career transition.

Curriculum details :-

=> Course Introduction :

- ~ course overview and dashboard description
- ~ Introduction of data science and its application in day to day life
- ~ Programming language overview
- ~ Installation (tools: sublime, vscode, pycharm, anaconda, atom, jupyter notebook, kite)
- ~ Virtual environment
- ~ Why python

=> Python Basic :

- ~ Introduction of python and comparison with other programming language
- ~ Installation of anaconda distribution and other python ide
- ~ Python objects, number & Booleans, strings
- ~ Container objects, mutability of objects
- ~ Operators - arithmetic, bitwise, comparison and assignment operators, operators precedence and associativity
- ~ Conditions (if else, if-elif-else), loops (while, for)
- ~ Break and continue statement and range function

=> String Objects :

- ~ basic data structure in python
- ~ String object basics
- ~ String inbuilt methods
- ~ Splitting and joining strings
- ~ String format functions

=> List Object Basics :

- ~ List methods
- ~ List as stack and queues
- ~ List comprehensions

=> Tuples, Sets, Dictionaries & its Function :

- ~ Dictionary object methods
- ~ Dictionary comprehensions
- ~ Dictionary view objects
- ~ Functions basics, parameter passing, iterators
- ~ Generator functions
- ~ Lambda functions
- ~ Map, reduce, filter functions

=> Memory Management :

- ~ Multithreading
- ~ Multiprocessing

=> OOPs Concepts :

- ~ oops basic concepts.
- ~ Creating classes
- ~ Pillars of oops
- ~ Inheritance
- ~ Polymorphism
- ~ Encapsulation
- ~ Abstraction
- ~ Decorator
- ~ Class methods and static methods
- ~ Special (magic/dunder) methods
- ~ Property decorators - getters, setters, and deletes

=> Files :

- ~ Working with files
- ~ Reading and writing files
- ~ Buffered read and write
- ~ Other file methods
- ~ Logging, debugger
- ~ Modules and import statements

=> Exception Handling and Difference between Exception and Error :

- ~ Exceptions handling with try-except
- ~ Custom exception handling
- ~ List of general use exception
- ~ Best practice exception handling

=> GUI Framework :

- ~ What is desktop and standalone application
- ~ Use of desktop app
- ~ Examples of desktop app
- ~ Tinker
- ~ Kivy

=> Database :

- ~ SQLite
- ~ MySQL
- ~ Mongo dB
- ~ NoSQL - Cassandra

=> Web API :

- ~ What is web API
- ~ Difference b/w API and web API
- ~ Rest and soap architecture
- ~ Restful services

=> Flask :

- ~ Flask introduction
- ~ Flask application
- ~ Open link flask
- ~ App routing flask
- ~ Url building flask
- ~ Http methods flask
- ~ Templates flask
- ~ Flask project: food app
- ~ Postman
- ~ Swagger

=> Django :

- ~ Django introduction
- ~ Django project: weather app
- ~ Django project: memes generator
- ~ Django project: blog app
- ~ Django project in cloud

=> Stream Lit :

- ~ Stream lit introduction
- ~ Stream lit project structure
- ~ Stream lit project in cloud

=> Pandas Basic :

- ~ Python pandas - series
- ~ Python pandas data frame
- ~ Python pandas panel
- ~ Python pandas - basic functionality
- ~ Reading data from different file system

=> Pandas Advance :

- ~ Python pandas re indexing python
- ~ Pandas iteration
- ~ Python pandas sorting.
- ~ Working with text data options & customization
- ~ Indexing & selecting
- ~ Data statistical functions
- ~ Python pandas - window functions
- ~ Python pandas - date functionality
- ~ Python pandas time delta
- ~ Python pandas - categorical data
- ~ Python pandas visualization
- ~ Python pandas - iotools

=> Dask :

- ~ Dask Array
- ~ Dask Bag
- ~ Dask DataFrame
- ~ Dask Delayed
- ~ Dask Futures
- ~ Dask API
- ~ Dask SCHEDULING
- ~ Dask Understanding Performance
- ~ Dask Visualize task graphs
- ~ Dask Diagnostics (local)
- ~ Dask Diagnostics (distributed)
- ~ Dask Debugging
- ~ Dask Ordering

=> Python Numpy :

- ~ Numpy - ND array object.
- ~ Numpy - data types.
- ~ Numpy - array attributes.
- ~ Numpy - array creation routines.
- ~ Numpy - array from existing.
- ~ Data array from numerical ranges.
- ~ Numpy - indexing & slicing.
- ~ Numpy advanced indexing.
- ~ Numpy broadcasting.
- ~ Numpy - iterating over array.
- ~ Numpy - array manipulation.
- ~ Numpy - binary operators.
- ~ Numpy - string functions.
- ~ Numpy - mathematical functions.
- ~ Numpy - arithmetic operations.
- ~ Numpy - statistical functions.
- ~ Sort, search & counting functions.

- ~ *Numpy - byte swapping.*
- ~ *Numpy - copies & views.*
- ~ *Numpy - matrix library.*
- ~ *Numpy - linear algebra*

=> Visualization :

- ~ *Matplotlib*
- ~ *Seaborn*
- ~ *Cufflinks*
- ~ *Plotly*
- ~ *Bokeh*

=> Statistics Basic :

- ~ *Introduction to basic statistics terms*
- ~ *Types of statistics*
- ~ *Types of data*
- ~ *Levels of measurement*
- ~ *Measures of central tendency*
- ~ *Measures of dispersion*
- ~ *Random variables*
- ~ *Set*
- ~ *Skewness*
- ~ *Covariance and correlation*

=> Probability Distribution Function :

- ~ *Probability density/distribution function*
- ~ *Types of the probability distribution*
- ~ *Binomial distribution*
- ~ *Poisson distribution*
- ~ *Normal distribution (Gaussian distribution)*
- ~ *Probability density function and mass function*
- ~ *Cumulative density function*
- ~ *Examples of normal distribution*
- ~ *Bernoulli distribution*
- ~ *Uniform distribution*
- ~ *Z stats*
- ~ *Central limit theorem*
- ~ *Estimation*

=> Statistics Advance :

- ~ *a Hypothesis*
- ~ *Hypothesis testings mechanism*
- ~ *P-value*
- ~ *T-stats*
- ~ *Student t distribution*
- ~ *T-stats vs. Z-stats: overview*
- ~ *When to use a t-tests vs. Z-tests*
- ~ *Type 1 & type 2 error*
- ~ *Bayes statistics (Bayes theorem)*
- ~ *Confidence interval(ci)*
- ~ *Confidence intervals and the margin of error*
- ~ *Interpreting confidence levels and confidence intervals*
- ~ *Chi-square test*
- ~ *Chi-square distribution using python*
- ~ *Chi-square for goodness of fit test*
- ~ *When to use which statistical distribution?*
- ~ *Analysis of variance (anova)*
- ~ *Assumptions to use anova*
- ~ *Anova three type*
- ~ *Partitioning of variance in the anova*
- ~ *Calculating using python*
- ~ *F-distribution*
- ~ *F-test (variance ratio test)*
- ~ *Determining the values of f*
- ~ *F distribution using python*

=> Linear Algebra :

- ~ *linear algebra*
- ~ *Vector*
- ~ *Scaler*
- ~ *Matrix*
- ~ *Matrix operations and manipulations*
- ~ *Dot product of two vectors*
- ~ *Transpose of a matrix*
- ~ *Linear independence of vectors*
- ~ *Rank of a matrix*
- ~ *Identity matrix or operator*
- ~ *Determinant of a matrix*
- ~ *Inverse of a matrix*
- ~ *Norm of a vector*
- ~ *Eigenvalues and eigenvectors*
- ~ *Calculus*

=> Solving Stats Problem with Python

=> Stats Problem Implementation with Spacy

=> Introduction to Machine Learning :

- ~ *AI vs ml vs dl vs ds*
- ~ *Supervised, unsupervised, semi-supervised, reinforcement learning*
- ~ *Train, test, validation split*

- ~ Performance
- ~ Overfitting, under fitting
- ~ Bias vs variance

=> Feature Engineering :

- ~ Handling missing data
- ~ Handling imbalanced data
- ~ Up-sampling
- ~ Down-sampling
- ~ Smote
- ~ Data interpolation
- ~ Handling outliers
- ~ Filter method
- ~ Wrapper method
- ~ Embedded methods
- ~ Feature scaling
- ~ Standardization
- ~ Mean normalization
- ~ Min-max scaling
- ~ Unit vector
- ~ Feature extraction
- ~ Pca (principle component analysis)
- ~ Data encoding
- ~ Nominal encoding
- ~ One hot encoding
- ~ One hot encoding with multiple categories
- ~ Mean encoding
- ~ Ordinal encoding
- ~ Label encoding
- ~ Target guided ordinal encoding
- ~ Covariance
- ~ Correlation check
- ~ Pearson correlation coefficient
- ~ Spearmans rank correlation
- ~ Vif

=> Feature Selection :

- ~ Feature selection
- ~ Recursive feature elimination
- ~ Backward elimination
- ~ Forward elimination

=> Exploratory Data Analysis :

- ~ Feature engineering and selection.
- ~ Analyzing bike sharing trends.
- ~ Analyzing movie reviews sentiment.
- ~ Customer segmentation and effective cross selling.
- ~ Analyzing wine types and quality.
- ~ Analyzing music trends and recommendations.
- ~ Forecasting stock and commodity prices

=> Regression :

- ~ Linear regression
- ~ Gradient descent
- ~ Multiple linear regression
- ~ Polynomial regression
- ~ R square and adjusted r square
- ~ Rmse , mse, mae comparison
- ~ Regularized linear models
- ~ Ridge regression
- ~ Lasso regression
- ~ Elastic net
- ~ Complete end-to-end project with deployment on cloud and ui

=> Logistics Regression :

- ~ Logistics regression in-depth intuition
- ~ In-depth mathematical intuition
- ~ In-depth geometrical intuition
- ~ Hyper parameter tuning
- ~ Grid search cv
- ~ Randomize search cv
- ~ Data leakage
- ~ Confusion matrix
- ~ Precision, recall, f1 score ,roc, auc
- ~ Best metric selection
- ~ Multiclass classification in lr
- ~ Complete end-to-end project with deployment in multi cloud platform

=> Decision Tree :

- ~ Decision tree classifier
- ~ In-depth mathematical intuition
- ~ In-depth geometrical intuition
- ~ Confusion matrix
- ~ Precision, recall, f1 score ,roc, auc
- ~ Best metric selection
- ~ Decision tree repressor
- ~ In-depth mathematical intuition
- ~ In-depth geometrical intuition
- ~ Performance metrics
- ~ Complete end-to-end project with deployment in multi cloud platform

=> Support Vector Machines :

- ~ Linear svm classification
- ~ In-depth mathematical intuition
- ~ In-depth geometrical intuition
- ~ Soft margin classification
- ~ Nonlinear svm classification
- ~ Polynomial kernel
- ~ Gaussian, rbf kernel
- ~ Data leakage
- ~ Confusion matrix
- ~ precision, recall, f1 score, roc, auc
- ~ Best metric selection
- ~ Svm regression
- ~ In-depth mathematical intuition
- ~ In-depth geometrical intuition
- ~ Complete end-to-end project with deployment

=> Naive Bayes :

- ~ Bayes theorem
- ~ Multinomial naive Bayes
- ~ Gaussian naive Bayes
- ~ Various type of Bayes theorem and its intuition
- ~ Confusion matrix
- ~ precision, recall, f1 score, roc, auc
- ~ Best metric selection
- ~ Complete end-to-end project with deployment

=> Ensemble Technique and its Types :

- ~ Definition of ensemble techniques
- ~ Bagging technique
- ~ Bootstrap aggregation
- ~ Random forest (bagging technique)
- ~ Random forest regressor
- ~ Random forest classifier
- ~ Complete end-to-end project with deployment

=> Boosting :

- ~ Boosting technique
- ~ Ada boost
- ~ Gradient boost
- ~ Xgboost
- ~ Complete end-to-end project with deployment

=> Stacking :

- ~ Stacking technique
- ~ Complete end-to-end project with deployment

=> KNN :

- ~ Knn classifier
- ~ Knn regressor
- ~ Variants of knn
- ~ Brute force knn
- ~ K-dimension tree
- ~ Ball tree
- ~ Complete end-to-end project with deployment

=> Dimensionality Reduction :

- ~ The curse of dimensionality
- ~ Dimensionality reduction technique
- ~ Pca (principle component analysis)
- ~ Mathematics behind pca
- ~ Scree plots
- ~ Eigen-decomposition approach

=> Clustering :

- ~ Clustering and their types
- ~ K-means clustering
- ~ K-means++
- ~ Batch k-means
- ~ Hierarchical clustering
- ~ DbSCAN
- ~ Evaluation of clustering
- ~ Homogeneity, completeness and v-measure
- ~ Silhouette coefficient
- ~ Davies-bouldin index
- ~ Contingency matrix
- ~ Pair confusion matrix
- ~ Extrinsic measure
- ~ Intrinsic measure
- ~ Complete end-to-end project with deployment

=> Anomaly Detection :

- ~ Anomaly detection types
- ~ Anomaly detection applications
- ~ Isolation forest anomaly detection algorithm
- ~ Isolation forest anomaly detection algorithm
- ~ Support vector machine anomaly detection algorithm
- ~ DbSCAN algorithm for anomaly detection
- ~ Complete end-to-end project with deployment

=> Time-Series :

- ~ What is a time series?
- ~ Old techniques
- ~ Arima
- ~ Acf and pacf
- ~ Time-dependent seasonal components.
- ~ Autoregressive (ar),
- ~ Moving average (ma) and mixed arma- modeler.
- ~ The random walk model.
- ~ Box-jenkins methodology.
- ~ Forecasts with arima and var models.
- ~ Dynamic models with time-shifted explanatory variables.
- ~ The koyck transformation.
- ~ Partial adjustment and adaptive expectation models.
- ~ Granger's causality tests.
- ~ Stationarity, unit roots and integration
- ~ Time series model performance
- ~ Various approach to solve time series problem
- ~ Complete end-to-end project with deployment
- ~ Prediction of nifty stock price and deployment

=> NLP Basic :

- ~ Tokenization
- ~ Pos tags and chunking
- ~ Stop words
- ~ Stemming and lemmatization
- ~ Named entity recognition (ner)
- ~ Word vectorization (word embedding)
- ~ Tfidf
- ~ Complete end-to-end project with deployment

=> Machine Learning Pipeline :

- ~ Aws segmaker
- ~ Aure ml studio
- ~ Ml flow
- ~ Kube flow

=> Model Retraining Approach

=> Auto ML :

- ~ H2o
- ~ Pycaret
- ~ Auto sklearn
- ~ Auto time series
- ~ Auto viml
- ~ Auto gluon
- ~ Auto viz
- ~ Tpot
- ~ Auto neuro

=> Neural Network A Simple perception :

- ~ Detail mathematical explanation
- ~ Neural network overview and its use case.
- ~ Various neural network architect overview.
- ~ Use case of neural network in nlp and computer vision.
- ~ Activation function -all name
- ~ Multilayer network.
- ~ Loss functions. - all 10
- ~ The learning mechanism.
- ~ Optimizers. - all 10
- ~ Forward and backward propagation.
- ~ Weight initialization technique
- ~ Vanishing gradient problem
- ~ Exploding gradient problem
- ~ Visualization of nn

=> Hardware Setup - GPU :

- ~ Gpu introduction.
- ~ Various type of gpu configuration.
- ~ Gpu provider and its pricing.
- ~ Paper space gpu setup.
- ~ Running model in gpu

=> Tensor Flow Installation Environment Setup For Deep Learning :

- ~ Colab pro setup
- ~ Tensor flow installation 2.0 .
- ~ Tensor flow installation 1.6 with virtual environment.
- ~ Tensor flow 2.0 function.
- ~ Tensor flow 2.0 neural network creation.
- ~ Tensor flow 1.6 functions.
- ~ Tensor flow 1.6 neural network and its functions.
- ~ Keras introduction.
- ~ Keras in-depth with neural network creation.
- ~ Mini project in tensorflow.
- ~ Tensorspace
- ~ Tensorboard integration
- ~ Tensorflow playground
- ~ Netron

=> Pytorch :

- ~ pytorch installation.
- ~ Pytorch functional overview.

~ *Pytorch neural network creation.*

=> **Mxnet :**

~ *Mxnet installation*
~ *Mxnet in depth function overview*
~ *Mxnet model creation and training*

=> **Keras Tuner :**

~ *Keras tuner installation and overview*
~ *Finding best parameter from keras tuner*
~ *Keras tuner application across various neural network*

=> **CNN Overview :**

~ *Cnn definition*
~ *Various cnn based architecture*
~ *Explanation end to end cnn network*
~ *Cnn explainer*
~ *Training cnn*
~ *Deployment in azure cloud*
~ *Performance tuning of cnn network*

=> **Advance Computer Vision - Part 1 :**

~ *Various cnn architecture with research paper and mathematics*
~ *Lenet-5 variants with research paper and practical*
~ *Alexnet variants with research paper and practical*
~ *Googlenet variants with research paper and practical*
~ *Transfer learning*
~ *Vggnet variants with research paper and practical*
~ *Resnet variants with research paper and practical*

=> **Advance Computer Vision - Part 2 :**

~ *Object detection in-depth*
~ *Transfer learning*
~ *Ssd with research paper and practical*

=> **Training of Custom Object Detection :**

~ *Tfod introduction*
~ *Environment setup with tfod*
~ *Gpu vs tpu vs cpu*
~ *Various gpu comparison*

=> **Advance Computer Vision - Part 3 :**

~ *Yolo v1 with research paper and practical*
~ *Yolo v2 with research paper and practical*
~ *Yolo v3 with research paper and practical*

=> **Object Segmentation :**

~ *Semantic segmentation*
~ *Masked rcnn*
~ *Practical with tfod*

=> **Object Tracking :**

~ *Detail of object tracking*
~ *Deep sort*
~ *Object tracking live project with live camera testing*

=> **OCR :**

~ *Introduction to ocr*
~ *Various framework and api for ocr*
~ *Practical implementation of ocr*

=> **Advance NLP with Deep Learning :**

~ *Overview computational linguistic.*
~ *History of nlp.*
~ *Why nlp*
~ *Use of nlp*

=> **Spacy :**

~ *Spacy overview.*
~ *Spacy function*
~ *Nltk*

=> **RNN :**

~ *Recurrent neural networks.*
~ *Long short term memory (lstm)*
~ *Bi lstm.*
~ *Stacked lstm*

=> **Word Embedding :**

~ *Word embedding*
~ *Word2vec*

=> **Attention Based Model :**

~ *Seq 2 seq.*
~ *Encoders and decoders.*
~ *Attention mechanism.*
~ *Attention neural networks*
~ *Self-attention*

=> **Transfer Learning in NLP :**

~ *Introduction to transformers.*
~ *Bert model.*
~ *Gpt1 model*
~ *Gpt2 model.*

=> Deployment of Model and Performance Tuning :

- ~ Deep learning model deployment strategies.
- ~ Deep learning project architecture
- ~ Deep learning model deployment phase.

=> Big Data Introduction :

- ~ What is big data?
- ~ Big data application
- ~ Big data pipeline

=> Hadoop :

- ~ Hadoop introduction
- ~ Hadoop setup and installation

=> Spark :

- ~ Spark
- ~ Spark overview.
- ~ Spark installation.
- ~ Spark rdd.
- ~ Spark data frame.
- ~ Spark architecture.
- ~ Spark deployment in local server

=> Kafka :

- ~ Kafka introduction
- ~ Kafka installation
- ~ Spark with Kafka

=> ML Ops :

- ~ Git

=> SQL :

- ~ Introduction
- ~ ER Daigram
- ~ Schema Design
- ~ Normalization
- ~ SQL SELECT Statement
- ~ SQL SELECT Using common functions
- ~ SQL JOIN Overview
- ~ INNER JOIN
- ~ LEFT JOIN
- ~ RIGHT JOIN
- ~ FULL JOIN
- ~ SQL Best Practice
- ~ INNER JOIN - Advanced
- ~ INNER JOIN & LEFT JOIN Combo
- ~ SELF JOIN
- ~ Joins & Aggregation - Subqueries
- ~ Sorting
- ~ Set Operations
- ~ SQL Views
- ~ Create a view
- ~ Create a view using DDL
- ~ SQL Insert - Advanced Technique
- ~ INSERT to create a table
- ~ INSERT new data to an existing table-1
- ~ INSERT new data to an existing table-2
- ~ INSERT new data to an existing table-3
- ~ INSERT new data to an existing table-4
- ~ SQL Update - Advanced Technique and TCL
- ~ SQL Aggregations

=> Advance Excel :

- ~ Microsoft Excel Fundamentals
- ~ Entering and Editing Text and Formulas
- ~ Working with Basic Excel Functions
- ~ Modifying an Excel Worksheet
- ~ Formatting Data in an Excel Worksheet
- ~ Inserting Images and Shapes into an Excel Worksheet
- ~ Creating Basic Charts in Excel
- ~ Printing an Excel Worksheet
- ~ Working with Excel Templates
- ~ Working with an Excel List
- ~ Excel List Functions
- ~ Excel Data Validation
- ~ Importing and Exporting Data
- ~ Excel PivotTables
- ~ Working with Excel's PowerPivot Tools
- ~ Working with Large Sets of Excel Data
- ~ Conditional Functions
- ~ Lookup Functions

=> Tableau :

- ~ Talking about Business Intelligence
- ~ Tools and Methodologies used in BI
- ~ Why Visualization is getting more popular
- ~ Why Tableau?
- ~ Gartner Magic Quadrant of Market Leaders
- ~ Future buisness impact of BI
- ~ Tableau Products

- ~ Tableau Architecture
- ~ Tableau Installation in local system
- ~ Introduction to Tableau Prep
- ~ Tableau Prep Builder User Interface
- ~ Data Preparation techniques using Tableau Prep Builder tool
- ~ How to connect Tableau with different data source
- ~ Visual Segments
- ~ Visual Analytics in depth
- ~ Filters, Parameters & Sets
- ~ Tableau Calculations using functions
- ~ Tableau Joins
- ~ Dynamic Dashboards and Stories

=> Power BI :

- ~ Power BI introduction and overview
- ~ Key Benefits of Power BI
- ~ Power BI Architecture
- ~ Power BI Process
- ~ Components of Power BI
- ~ Power BI - Building Blocks
- ~ Power BI vs other BI tools
- ~ Power Installation
- ~ Overview of Power BI Desktop
- ~ Data Sources in Power BI Desktop
- ~ Connecting to a data Sources
- ~ Query Editor in Power BI
- ~ Views in Power BI
- ~ Field Pane
- ~ Visual Pane
- ~ Custom Visual Option
- ~ Filters
- ~ Introduction to using Excel data in Power BI
- ~ Import Power View and Power Pivot to Power BI
- ~ Power BI Publisher for Excel
- ~ Introducing Power BI Mobile
- ~ Power Query Introduction
- ~ Query Editor Interface
- ~ Clean and Transform your data with Query Editor
- ~ Data Type
- ~ Column Transformations vs Adding Columns
- ~ Text Transformations
- ~ Data Modelling
- ~ Calculated Columns

=> GAN

Project details :-

=> Python Project :

- ~ Weeding script
- ~ Image resizing
- ~ Web scrapping
- ~ Web crawlers for image data sentiment analysis and product review sentiment analysis.
- ~ Integration with web portal.
- ~ Integration with rest api, web portal and mongo db. on azure
- ~ Deployment on web portal on azure.

=> Major Projects :

- ~ Healthcare analytics prediction of medicines based on Fitbit band.
- ~ Revenue forecasting for startups.
- ~ Prediction of order cancellation at the time of ordering inventories.
- ~ anomaly detection in inventory packaged material.
- ~ Fault detection in wafers based on sensor data.
- ~ Demand forecasting for fmcg product.
- ~ Threat identification in security system.
- ~ Defect detection in vehicle engine.
- ~ Food price forecasting with zomato dataset.
- ~ Fault detection in wafers based on sensor data.
- ~ Cement strength reg.
- ~ Credit card fraud.
- ~ Forest cover classification.
- ~ Fraud detection.
- ~ Income prediction.
- ~ Mushroom classifier.
- ~ phishing classifier
- ~ Thyroid detection.
- ~ Visibility climate

=> Computer Vision Project :

- ~ Traffic surveillance system.
- ~ Object identification.
- ~ Object classification.
- ~ Tensorflow object detection.

=> Mini NLP Project :

- ~ Machine translation.
- ~ Abstractive text summarization.
- ~ Keyword spotting.
- ~ Language modelling.

=> NLP Transfer Learning Project :

- ~ *Deployment and integration with UI machine translation.*
- ~ *Question answering (like chat bot)*
- ~ *Sentiment analysis imdb.*
- ~ *Text search (with synonyms).*

=> NLP End to End Project with Architecture and Deployment :

- ~ *Movie review using bert*
- ~ *NER using Bert*
- ~ *Pos bert*
- ~ *Text generation gpt 2*

=> NLP Project End to End with Deployment in Various Cloud and UI Integration :

- ~ *Topic modeling.*
- ~ *Text to speech*

=> Tableau Project :

- ~ *Human Resource - Tableau*

=> Power BI Project :

- ~ *Retail Insights- Power BI*

TypeScript Crash Course

Topic Name : WEB DEVELOPEMENT

Sub-topic Name : JAVASCRIPT

Course link : <https://ineuron.ai/course/TypeScript-Crash-Course>

Course Description :-

Grab the fundamentals of TypeScript through this crash course. This course helps you to get started with Typescript.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => Basics of Typescript
- => Installation of TypeScript

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Instructors :-

=> Hitesh Choudhary :

~ I like to make videos related to code and tech in my free time. I also lead a few tech teams in startups, help in hiring talent for companies. I am also on a part time traveller, with 31 countries checked off so far!

Curriculum details :-

- => TypeScript :
 - ~ Why to learn Typescript
 - ~ Typescript is not what you think
 - ~ How to install Typescript
 - ~ Your first intro to Typescript docs
 - ~ Number, Boolean & Type Inference
 - ~ Don't use ANY in Typescript

Vedic Math

Topic Name : K12

Sub-topic Name : CLASS8

Course link : <https://ineuron.ai/course/Vedic-Math>

Course Description :-

This course will help you solve complex mathematical problems using Vedic Mathematics. This course is curated on a set of concepts that will help you improve your calculations to an extent where before you pick up a pen, you would find the answers by simplifying calculations into simple steps. If you despise numbers, This course will help you interactively appreciate the beauty of mathematics.

Course Features :-

- => Online Instructor-led learning
- => Practical Implementation
- => Integrate academic knowledge with the tech
- => Real-time Project
- => Live Class Recording
- => Doubt Clearing
- => Assignment in all the Module
- => Quiz in every Module
- => Career Counselling
- => Completion Certificate

What you will learn :-

- => Introduction about Vedic math's
- => Benefits of Vedic math's
- => Addition of numbers
- => Subtraction of numbers
- => Multiplication of numbers
- => Division of numbers
- => Square of a number
- => Cube of a number
- => Square root of a number
- => Cube root of a number

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Curriculum details :-

=> Introduction to VEDIC MATHS :

- ~ What is VEDIC MATHS?
- ~ Benefits of Vedic Maths
- ~ Why we learn VEDIC MATHS ?

=> Basics VEDIC MATHS :

- ~ Find the complement of 1 digit number/ 2 digit number/ 3 digit number/ any digit number?
- ~ Tables for 9/19/29/39/129/149/.
- ~ All from 9 and the Last From 10
- ~ Multiplication of 2 numbers where number of digits are same in both number and sum of unit digit number is 10
- ~ Multiplication of 2 numbers where digits are same in both numbers except unit digit number
- ~ Multiplication with 11
- ~ Multiplication with 12
- ~ Universal multiplication like multiplication for following exs:- 1) $2^2 \times 2$ 2) $3^3 \times 3$ 3) $4^4 \times 4$ 4) $2^3 \times 5$ 5) $3^4 \times 6$ 6) $2^4 \times 7$ 7) $3^2 \times 2$ etc (called as URDHVA TIRYA GAMYAM)
- ~ Square of different types of number (For example, whose unit digit /the last digit is 1/5/6/4
- ~ Cubes of 2 digit number

=> Intermediate :

- ~ Division of any number by 5, 8 & 98
- ~ Division of any number by 11
- ~ Division of any number by 12-19
- ~ Division of any number by 25, 50 and 100

- ~ Division by factors
- ~ Percentages
- ~ Addition of odd, even series of numbers
- ~ Multiplication of numbers ending with 5
- ~ Multiplication of numbers with 15
- ~ Multiplication of whole number with mixed fractions

=> Advanced VEDIC MATHS :

- ~ Division of whole number with mixed fractions
- ~ Addition of special fractions
- ~ Square of any number nearer to base
- ~ Multiplications with 9/99/999 when
- 1) Multipliers are same digit
- 2) Less digits than multiplier
- 3) More digits than multiplier
- ~ Different types of base multiplication for ex. $97 * 94$ (less than base 100), $14 * 12$
- ~ Multiplication of numbers like $(42 * 46)$ considering primary & secondary base
- ~ different (in this ex, 10 & 50 are two different base)
- ~ Cubes of numbers closer to bases
- ~ Division (Nikhilam method where divider is less than base number or nearer to base number)
- ~ Quickest division by 9,99,999,9999
- ~ Repeated digit base number squares
- ~ Vinculum of number at Unit and tens places
- ~ Squares by duplex method
- ~ Square root
- ~ Cube root

=> Application: :

~ We will make a UI where we will provide a set of questions where kids can answer those questions & check if it is correct or not/ Similarly, In the backend, they can do logic according to Vedic formula.
Plus, we will provide a hint that explains the Vedic formula/procedure to find the solution in the front end.

Apache Druid

Topic Name : BIG DATA

Sub-topic Name : TECH STACK

Course link : <https://ineuron.ai/course/Apache-Druid>

Course Description :-

Apache Druid is a real-time database to power modern analytics applications. It is a modern technology which has made itself indispensable in the world of big data and data analytics. It comes up with cutting-edge features like easy integration in data pipelines, fast consistent queries, high concurrency support, support in both on-prem and cloud infrastructure, and many more.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => Introduces Apache Druid
- => Apache Druid Installation
- => Walkthrough of Apache Druid Console
- => Data Transformation while ingestion
- => Data filtering while ingestion
- => Nested data parsing while ingestion
- => Data Rollup while ingestion
- => Introduction to Data Segments
- => Data Deletion in Druid
- => Other ways of loading data into Druid
- => Querying data in Druid
- => Druid Internals
- => Druid integration with Superset
- => Druid at Netflix

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Instructors :-

=> Shruti Mantri :

~ Shruti Mantri is a well-known software architect, instructor and mentor in the industry. She has 10+ years of experience in the software industry, and has worked with different organizations like Oracle, Flipkart, Amazon, Myntra and Twitter. She is known for her expertise in the data engineering field, and has a sound knowledge on the latest technologies in this domain. She has helped develop data platform at organizations, and guided several mentees in understanding data engineering and how to get better at it.

Curriculum details :-

- => Introduces Apache Druid :
 - ~ An Introduction to Apache Druid
 - ~ Course Objectives
 - ~ Where to use & where not to use Druid
- => Apache Druid Installation :
 - ~ Single Server Installation
 - ~ Overview of the Druid Console
- => Understanding Data Load :
 - ~ Load data from file using console
 - ~ Other ways to load data from file
 - ~ Transformations over data
 - ~ Applying filters over data

- ~ *Parsing nested data*
- ~ *Rollup Data*

=> Other Data Loads :

- ~ *Load data from Kafka*
- ~ *Load data from Https*
- ~ *Load ORC data format*

=> Querying :

- ~ *Druid SQL Queries*
- ~ *Native Queries*
- ~ *Querying Data via Http*

=> Druid Internals :

- ~ *Druid Architecture*

=> Use Cases :

- ~ *Integrating Superset with Druid*
- ~ *Industry Use-case: Druid at Netflix*

Tibco Spotfire

Topic Name : DATA ANALYTICS

Sub-topic Name : DASHBOARDING

Course link : <https://ineuron.ai/course/Tibco-Spotfire>

Course Description :-

Upskill your analytics skills through the power of Tibco Platform. Tibco spotfire is an analytical platform enabling us to explore and visualize new discoveries in data through immersive dashboards and advanced analytics. You will be able to enhance your skill by exploring this Tibco Analytical platform.

Course Features :-

- => Learning of different tibco spotfire Services
- => Understanding Data integration
- => BI & Analytics
- => Creating charts, reports, graphs, dashboards for analytical purpose
- => Completion Certificate
- => Understanding spotfire Features

What you will learn :-

- => Tibco spotfire Features
- => Analytics
- => spotfire mods
- => Dashboard
- => creating report
- => creating Dashboard

Requirements :-

- => No prior knowledge in Analytics
- => System with Internet Connection
- => Interest to learn
- => Basic knowledge of BI
- => Dedication

Instructors :-

=> Khushali Shah :

~ A data scientist having rich experience working with MNCs and start-ups in the field of data science and machine learning. She has expertise in Chatbot development for various domains & been developing professionally for 6+ years with diverse job history. She also had positions in software module development, web app development, functional designs, requirement gathering, client interaction, and server setup/admin & can help everywhere in the stack; she loves wearing multiple hats to an extent. She also believes in enhancing her skills by training and learning new things day by day.

Curriculum details :-

- => Tibco Fundamentals :
 - ~ Overview Preview
 - ~ Tibco feature Preview
 - ~ Tibco benefits features
 - ~ Tibco dashboard overview
 - ~ Tibco spotfire mods
 - ~ Tibco spotfire analysis
 - ~ Tibco spotfire handson

Machine Learning Masters

Topic Name : DATA SCIENCE

Sub-topic Name : MACHINE LEARNING

Course link : <https://ineuron.ai/course/Machine-Learning-Masters>

Course Description :-

Machine Learning Masters

Course Features :-

- => Machine Learning in depth from beginning to advance discussion and implementation with Deployment.
- => Deep learning in-depth topic wise discussion and implementation with the project.
- => Docker and Kubernetes end to end with CI/CD pipeline for machine learning.
- => End to End Model Deployment in Azure, GCP, AWS, and Pivotal Cloud.
- => Python spark implementation with the project.
- => Time Series end to end implementation in machine learning and deep learning.
- => 26 + hands-on industry real-time projects.
- => Power BI and Tableau self-placed course.
- => Machine Learning Deep Learning Masters Certificate
- => 200 hours live interactive classes.
- => Every week doubt clearing session after the live classes.
- => Lifetime Dashboard access.
- => Doubt clearing one to one
- => Doubt clearing through mail and support team
- => Assignment in all the module
- => 20+ use case of Machine learning
- => A live project with real-time implementation
- => Resume building
- => career guidance
- => interview Preparation
- => Regular assessment
- => Job alerts
- => Online Instructor-led learning: Live teaching by instructors
- => Product Demo

What you will learn :-

- => Python
- => Stats
- => Machine learning
- => Deep learning
- => Data analytics
- => Mock interview
- => Interview preparation
- => Resume building

Requirements :-

- => Dedication
- => Laptop with internet connectivity

Instructors :-

=> Sunny Bhavleen Chandra :

~ Sr. Data Scientist and lecturer at iNeuron.ai with working experience in computer vision, natural language processing and embedded systems. Hands-on experience leveraging machine learning, deep learning, transfer learning models to solve challenging business problems. Also, he has a vast interest in Robotics.

=> Sourangshu Pal :

~ Visual Computing Engineer and instructor at iNeuron.ai having 3 years of diverse experience in the discipline of visual computing with specialization in Deep Learning and Computer Graphics. Loves to analyze, process, and model visual data then interpret the insights to create actionable plans for solving

challenging business problems.

=> krish naik :

~ Having 10+ years of experience in Data Science and Analytics with product architecture design and delivery. Worked in various product and service based Company. Having an experience of 5+ years in educating people and helping them to make a career transition.

=> Sudhanshu Kumar :

~ Having 8+ years of experience in Big data, Data Science and Analytics with product architecture design and delivery. Worked in various product and service based Company. Having an experience of 5+ years in educating people and helping them to make a career transition.

Curriculum details :-

=> Course Introduction :

- ~ Introduction of Data science and its application in Day to Day life Preview
- ~ Course overview and Dashboard description Preview

=> Python Core :

- ~ Introduction of python and comparison with other
- ~ Programming language
- ~ Installation of Anaconda Distribution and other python
- ~ IDE Python Objects, Number & Booleans, Strings
- ~ Container objects, Mutability of objects
- ~ Operators Arithmetic, Bitwise, Comparison and Assignment operators, Operators Precedence and associativity
- ~ Conditions(If else, if elif else) Loops(While ,for)
- ~ Break and Continue statement and Range Function.

=> String Objects and collections :

- ~ String object basics
- ~ String methods
- ~ Splitting and Joining Strings
- ~ String format functions
- ~ List object basics
- ~ List as stack and Queues
- ~ List comprehensions

=> Tuples, Set , Dictionaries Functions :

- ~ Tuples, Sets Dictionary Object basics, Dictionary Object methods, Dictionary View Objects.
- ~ Functions basics, Parameter passing, Iterators Generator functions
- ~ Lambda functions
- ~ Map , Reduce, Filter functions

=> OOPS concepts Working with Files :

- ~ OOPS basic concepts
- ~ Creating classes and Objects Inheritance
- ~ Multiple Inheritance
- ~ Working with files
- ~ Reading and writing files
- ~ Buffered read and write
- ~ Other File methods

=> Exception Handling :

- ~ Exceptions Handling with Try except

=> Api :

- ~ Flask introduction
- ~ Flask Application
- ~ Open link Flask
- ~ App Routing Flask
- ~ URL Building Flask
- ~ HTTP Methods Flask

=> Database :

- ~ Mongo DB SQL
- ~ Lite python SQL

=> Python pandas Modules :

- ~ Python Pandas Series
- ~ Python Pandas DataFrame
- ~ Python Pandas Panel
- ~ Python Pandas Basic functionality

=> Python Numpy :

- ~ NumPy Narray Object
- ~ NumPy Data Types
- ~ NumPy Array Attributes
- ~ NumPy Array Creation Routines
- ~ NumPy Array from Existing
- ~ Data Array From Numerical Ranges
- ~ NumPy Indexing & Slicing
- ~ NumPy Advanced Indexing
- ~ NumPy Broadcasting
- ~ NumPy Iterating Over Array
- ~ NumPy Array Manipulation
- ~ NumPy Binary Operators
- ~ NumPy String Functions
- ~ NumPy Mathematical Functions
- ~ NumPy Arithmetic Operations
- ~ NumPy Statistical Functions
- ~ Sort , Search & Counting Functions
- ~ NumPy Byte Swapping
- ~ NumPy Copies Views
- ~ NumPy Matrix Library

~ NumPy Linear Algebra

=> Exploratory Data Analysis :

- ~ Feature Engineering and Selection
- ~ Building Tuning and Deploying Models
- ~ Analyzing Bike Sharing Trends
- ~ Analyzing Movie Reviews Sentiment
- ~ Customer Segmentation and Effective Cross Selling
- ~ Analyzing Wine Types and Quality
- ~ Analyzing Music Trends and Recommendations
- ~ Forecasting Stock and Commodity Prices

=> Statistics :

- ~ Descriptive Statistics
- ~ Sample vs Population statistics Random Variables
- ~ Probability distribution function Expected value
- ~ Binomial Distribution
- ~ Normal Distribution z score
- ~ Central limit Theorem
- ~ Hypothesis testing Z Stats vs T stats
- ~ Type 1 type 2 error
- ~ Confidence interval
- ~ Chi Square test
- ~ ANOVA test
- ~ F stats

=> Machine Learning 1 :

- ~ Introduction
- ~ Supervised , Unsupervised, Semi supervised, Reinforcement Train , Test, Validation Split
- ~ Performance Overfitting , underfitting OLS.
- ~ Linear Regression assumption.
- ~ R square adjusted
- ~ R square Intro to Scikit learn
- ~ Training methodology
- ~ Hands on linear regression
- ~ Ridge Regression
- ~ Logistics regression
- ~ Precision Recall ROC curve
- ~ F Score

=> Machine Learning 2 :

- ~ Decision Tree Cross
- ~ Validation Bias vs Variance
- ~ Ensemble approach Bagging
- ~ Boosting Random
- ~ Forest Variable Importance

=> Machine Learning 3 :

- ~ XGBoost
- ~ Hands on XgBoost
- ~ K Nearest Neighbour
- ~ Lazy learners
- ~ Curse of Dimensionality
- ~ K NN Issues
- ~ Hierarchical clustering K Means
- ~ Performance measurement
- ~ Principal Component analysis
- ~ Dimensionality reduction
- ~ Factor Analysis

=> Machine Learning4 :

- ~ SVR
- ~ S V M
- ~ Polynomial Regression
- ~ Ada boost
- ~ Gradient boost
- ~ Gaussian mixture
- ~ Anomaly detection
- ~ Novelty detection algorithm Stacking
- ~ K NN regressor
- ~ Decision tree regressor DBSCAN

=> Natural Language Processing :

- ~ Text Analytics
- ~ Tokenizing , Chunking
- ~ Document term
- ~ Matrix TFIDF
- ~ Sentiment analysis hands on

=> Spark :

- ~ Spark overview.
- ~ Spark installation.
- ~ Spark RDD.
- ~ Spark dataframe .
- ~ Spark Architecture.
- ~ Spark ML lib.
- ~ Spark Nlp
- ~ Spark linear regression.
- ~ Spark logistic regression.
- ~ Spark Decision Tree.
- ~ Spark Naive Bayes

- ~ Spark xg boost
- ~ Spark time series.
- ~ Spark Deployment in local server
- ~ Spark job automation with scheduler.

=> Deep Learning :

- ~ Deep Learning Introduction.
- ~ Neural Network Architecture.
- ~ Loss Function.
- ~ Cost Function.
- ~ Optimizers.
- ~ CNN architecture.
- ~ Build First Classifier in CNN.
- ~ Deploy Classifier over cloud.
- ~ RNN overview.
- ~ GRU.
- ~ LSTM.
- ~ Time Series using RNN LSTM.
- ~ Customer Feedback analysis using RNN LSTM.

=> Time Series :

- ~ Arima
- ~ Sarima .
- ~ Auto Arima
- ~ Time series using RNN LSTM .
- ~ Prediction of NIFTY stock price.

=> Deployment :

- ~ Deployment of all the project In cloudfoundary , AWS AZURE and Google cloud platform
- ~ Expose api to web browser and mobile application retraining a pproach of Machine learning model
- ~ Devops infrastructure for machine learning model
- ~ Data base integration and scheduling of machine learning model and retraining c ustom machine learning training approach.
- ~ AUTO ML
- ~ Discussion on infra cost and data volume
- ~ P rediction based on streaming data

=> Extra session :

- ~ Discussion on project explanation in interview
- ~ Data scientist roles and responsibilities
- ~ Data scientist day to day work
- ~ Companies which hire a data scientist
- ~ Resume discussion with our team one to one

=> Tableau and power Bi self placed session :

- ~ Business Intelligence (BI) Concepts.
- ~ Microsoft Power BI (MSPBI) introduction.
- ~ Connecting Power BI with Different Data sources.
- ~ Power Query for Data Transformation.
- ~ Data Modelling in Power BI.
- ~ Reports in Power BI Reports and Visualisation types in Power BI.
- ~ Dashboards in Power BI.
- ~ Data Refresh in Power BI.
- ~ Traditional Visualisation(Excel) vs Tableau.
- ~ About Tableau.
- ~ Tableau vs Other BI Tool Pricing.

=> Tableau Interview Questions.

Project details :-

=> Python project :

- ~ Web crawlers for image data sentiment analysis and product review sentiment analysis
- ~ Integration with web portal
- ~ Integration with rest a A pi W eb portal and Mongo DB on Azure
- ~ Deployment on web portal on Azure
- ~ Text mining
- ~ Social media data churn

=> Chatbot Project :

- ~ Chatbot using Microsoft Luis
- ~ Chatbot using google Dialog flow
- ~ Chatbot using Amazon Lex
- ~ Chatbot using Rasa NLU
- ~ Deployemnt of chatbot with web , Telegram , Whatsapp , Skype

=> Machine learning project :

- ~ Healthcare analytics prediction of medicines based on FIT BITband
- ~ Revenue forecasting for startups
- ~ Prediction of order cancellation at the time of ordering inventories.
- ~ Anamoly detection in inventory packaged material.
- ~ Fault detection in wafferes based on sensordata
- ~ Demand forecasting for FMCG product.
- ~ Threat identification in security system.
- ~ Defect detection in vehicle engine.
- ~ Food price forecasting with Zomato dataset.
- ~ Fault detection in wafferes based on sensor data.
- ~ Cement_Strength _ reg.
- ~ Credit Card Fraud.
- ~ Forest_Cover_Classification .
- ~ Fraud Detection.
- ~ Income Prediction.

- ~ *Mushroom classifier., Phising Classifier , Thyroid_Detection .*
- ~ *Visibility climate.*

=> Deep Learning projects :

- ~ *Customer Feedback analysis using RNN LSTM.*
- ~ *Family member detection.*
- ~ *Industry financial growth prediction.*
- ~ *Speech recognition based attendance system.*
- ~ *Vehicle Number plate detection and recognition system.*

=> Tableau and power Bi Projects :

- ~ *Project 1. Project Sales.*
- ~ *Project 2. Financial Report.*
- ~ *Project 3. HealthCare.*
- ~ *Project 4. Procurement Spend Analysis.*
- ~ *Project 5. Human Resource Tableau*

Pro Live Classes

Sub-topic Name : Null

Course link : <https://ineuron.ai/course/Pro-Live-Classes>

Course Description :-

Pro Live Classes are designed and taught by industry expert specializing in various domains and sharing their experience to our students.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises

What you will learn :-

- => Developing and Open Sourcing an ML/DL package

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Curriculum details :-

- => Day1 :
 - ~ *Developing and Open Sourcing an ML/DL package*
- => Day 2 :
 - ~ *Dynamic Programming*
- => Day 3 :
 - ~ *Path To Be A Data Engineer*

Game Development using Unity

Topic Name : K12

Sub-topic Name : CLASS10

Course link : <https://ineuron.ai/course/Game-Development-using-Unity>

Course Description :-

This course provides an overview of the core principles of gaming using Unity. You will learn to define a game, as well as the mechanics and rules that govern a variety of games. After finishing this course, you will have a thorough understanding of the entire game development process.

Course Features :-

- => Online Instructor-led learning
- => Practical Implementation
- => Integrate academic knowledge with the tech
- => Real-time Project
- => Live Class Recording
- => Doubt Clearing
- => Assignment in all the Module
- => Quiz in every Module
- => Career Counselling
- => Completion Certificate

What you will learn :-

- => Introduction using Unity
- => Introduction to game engine
- => Introduction to C# programming
- => 2D physics concepts
- => Introduction to animation
- => 3D game development
- => Publishing games to various platforms

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Curriculum details :-

- => Introduction to the course :
 - ~ What is Unity?
 - ~ What do you learn in Unity?
 - ~ Why Unity?
 - ~ Downloading Unity and installation
 - ~ Unity Objects
 - ~ Unity Components
 - ~ Creating your own components
- => Assignment no.1 :
 - ~ Write down the Unity Competitors and alternatives?
 - ~ Are there free gaming softwares available? if yes please name them?
- => Game Engine :
 - ~ Engine concepts
 - ~ Development tools
 - ~ User interface text
 - ~ Countdown timer
 - ~ Digital clock
 - ~ Sprites
- => C# programming :
 - ~ What is scripting?
 - ~ What is c# language?
 - ~ Introduction to C# language
 - ~ C# coding fundamentals
 - ~ Loops and functions
 - ~ If - Else statements

=> 2D Physics concepts :

- ~ Rigidbody components
- ~ Unity colliders
- ~ Physics materials
- ~ scripting collision events
- ~ Importing asset
- ~ Making character move
- ~ Giving jumping abilities

=> Animation :

- ~ Simple Unity animation
- ~ Animator states
- ~ Scripting animations
- ~ Animations and colliders

=> Assignment no.2 :

- ~ Create your own Avatar

=> Sound effects :

- ~ Sound files
- ~ Adding sounds to game objects
- ~ Scripting sounds

=> Assignment no.3 :

- ~ Create a cloud object and give a thunder cloud sound effect

=> 3D game development :

- ~ Creating multiple scenes
- ~ Creating lighting effects
- ~ Creating Controlling character script
- ~ Controlling camera movements
- ~ Implementation of Occlusion culling

=> Assignment no.4 :

~ Create your own game where a rabbit is walking in the farm, if he digs and finds a carrot, give a point and if he digs and finds nothing, decrease the point.

=> Artificial Intelligence :

- ~ What is Artificial intelligence?
- ~ Artificial intelligence concepts
- ~ Flowcharts and algorithms
- ~ Scripting AI

=> Publishing Games :

- ~ Publishing games to PC
- ~ Publishing games to MAC
- ~ Publishing games to LINUX
- ~ Publishing games to Smartphones
- ~ Publishing games to Game Consoles

=> Project :

- ~ Create a game where a boy is riding a cycle on road

Java Coding Interview Preparation

Topic Name : PROGRAMMING

Sub-topic Name : JAVA

Course link : <https://ineuron.ai/course/Java-Coding-Interview-Preparation>

Course Description :-

This course is designed mostly for Java test takers.

Course Features :-

=> Quizzes

=> Course completion certificate

What you will learn :-

=> Java Theoretical Test

=> Java Practical Test

Requirements :-

=> System with minimum i3 processor or better

=> At least 4 GB of RAM

=> Working internet connection

=> Dedication to solve

Curriculum details :-

=> Java Test :

- ~ *Java Test 1*
- ~ *Java Test 2*
- ~ *Java Test 3*
- ~ *Java Test 4*
- ~ *Java Test 5*
- ~ *Java Test 6*
- ~ *Java Test 7*
- ~ *Java Test 8*
- ~ *Java Test 9*
- ~ *Java Test 10*
- ~ *Java Test 11*
- ~ *Java Test 12*
- ~ *Java Test 13*
- ~ *Java Test 14*
- ~ *Java Test 15*
- ~ *Java Test 16*
- ~ *Java Test 17*
- ~ *Java Test 18*
- ~ *Java Test 19*
- ~ *Java Test 20*

Data Structure and Algorithm Job Preparation

Topic Name : DATA STRUCTURE

Sub-topic Name : DSA INTERVIEW

Course link : <https://ineuron.ai/course/Data-Structure-and-Algorithm-Job-Preparation>

Course Description :-

Algorithmic programming techniques are a must-have skill. Learn Algorithms through programming and puzzle solving to advance your Software Engineering or Data Science career. Then, implement each algorithmic problem in this program to ace coding interviews.

Course Features :-

- => Downloadable resources
- => Roadmap
- => Assignments
- => Quizzes
- => Interview questions
- => Completion certificate

What you will learn :-

- => Problem solving
- => Analytical skill
- => Design Solution
- => Architecture design
- => Answer Confidently in interview.
- => Upscale your skill as a Developer.

Requirements :-

- => Prior Knowledge of Data Structure & Algorithms concepts.
- => A System with internet connection.
- => Your dedication

Instructors :-

- => Priya Bhatia :
 - ~ Expertise in data structure competitive programming and solving analytical problems and implementing data structure algorithm in multiple programming language. I have done my M.Tech in Artificial Intelligence at IIT Hyderabad and have an experience of implementation in multiple projects.

Curriculum details :-

- => Maximum Subarray Sum :
 - ~ Given an integer array, find the sum of the largest contiguous subarray within the array. Note that if all the elements are negative, you should return 0. [Preview](#)
- => K Closest Points :
 - ~ Given a list of coordinates, write a function to find k closest points(measured by Euclidean Distance) to the origin. [Preview](#)
- => Finding of Kth Smallest Element in 2D Sorted Matrix :
 - ~ Given an n-by-n matrix of elements that are sorted in ascending order both in the columns and rows of the matrix. Return the kth smallest element of the matrix.
- => Mirror Image in Binary Tree :
 - ~ Given a binary tree, write a function to determine whether the tree is a mirror image of itself. Two trees are mirror images of each other if their root values are the same, and the left subtree is a mirror image of the right subtree.
- => Maximum Product :
 - ~ Given an integer array, return the maximum product of any three numbers in the array. [Preview](#)
- => Intersection Of Elements in an array :
 - ~ Given two arrays, write a function to get the intersection of elements between the two arrays.
- => Diameter Of Tree :
 - ~ Given a binary tree, write a function to determine the diameter of the tree, which is the longest path between any two nodes.
- => Maximum Length Of Common Subarray :
 - ~ Given two arrays, return the maximum length of the common subarray within both arrays. For example, if two arrays are [2, 5, 7, 9] and [1, 2, 5, 8], then the function return two as the length of the maximum common subarray is [2,5].
- => Finding of Peak Elements :
 - ~ Given an integer array, find the peak element and return its index. Here, Peak Element is an element that is strictly greater than its neighbours. If an array contains multiple peaks, return the index to any of the peaks.
- => Top K Frequent Elements :
 - ~ Given an integer array and integer k, return the k most frequent elements. For example, if an array is [2, 2, 3, 3, 3, 5] and k = 2 then function return [2, 3].

=> Permutation Of list :

~ Given a list of one or more distinct integers, write a function to generate all the permutations of those integers.

=> Combinations of k numbers :

~ Given an integer n and an integer k , output a list of all the combinations of k numbers chosen from 1 to n . For example, if $n = 3$ and $k = 2$ function should return $[1,2],[1,3],[2,3]$.

=> Removal Of kth Node from end :

~ Given a linked list, return the head of the same linked list but with the k th node from the end of the linked list removed. For example given the linked list 2->3->7->1->4 and $k = 3$ then function remove the 7 node and return the linked list 2->3->1->4.

=> Length Of Longest Path :

~ Given the m -by- n matrix with positive integers, find the length of the longest path of increasing integers with the matrix. For example if the matrix is $\begin{bmatrix} 1, & 2, & 3 \\ 4, & 5, & 6 \\ 7, & 8, & 9 \end{bmatrix}$ then function should return 5 since the longest path would be 1-2-5-6-9

=> Reverse Linked List from the given start and end position :

~ Given the head of the singly linked list and two integer start and end where $start \leq end$, reverse the nodes of the list from position start to position end and return the reversed linked list.

=> Finding of Cycle in Linked List :

~ Given the head of a linked list, determine if the linked list has a cycle in it or not.

=> Rotate Array :

~ Given an array, rotate the array to the right by k steps where k is non-negative.

=> Longest Substring Without Repeating Characters :

~ Given a string s , find the length of the longest substring without repeating characters. For example, if $s = "abcdabc"$ the result should be "abcd" with the length of 4.

=> Number of Friend Group :

~ Say that there are n people. If person X is a friend with person Y , and person Y is a friend with person Z , then person X is considered to be an indirect friend of person Z . Define a friend group to be any group that is either direct or indirect friends. Given an n -by- n

~ Adjacency matrix N where $N[i][j]$ is one of the persons i and person j are friends and is zero otherwise, write a function to get how many friend groups exist.

=> Greedy Algorithms vs Dynamic Programming :

~ As you are aware of the fact that both Greedy Algorithms and Dynamic Programming are used to solve the optimization problem, then what's the difference between these two approaches and how do we decide when we should go for what approach?

=> QuickSort :

~ In QuickSort, sorting of n numbers, $n/10$ th element is selected as Pivot using $O(\log N)$. Then, what will be the worst-case time complexity of QuickSort?

=> Correlation :

~ Given two lists X and Y , return their correlation.

=> QuickSort vs MergeSort :

~ Out of MergeSort and QuickSort, which one do you think is more suitable for practical use cases and why?

~ Why is QuickSort preferred for Array and MergeSort for LinkedList?

=> Median Calculation :

~ Given a continuous stream of integers, write a class with functions to add new integers to the stream and a function to calculate the median at any time.

=> Length Of longest well-formed substring :

~ Given a string with left and right parentheses, write a function to determine the length of the longest well-formed substring. For example, if the input string is " $()()()$ " then the function should return four since the longest well-formed substring is " $()()()$ ".

=> Sum to the target Number :

~ Given a target number, generate a random sample of integers that sum to that target that are also within a standard deviation of the mean.

=> Sort an array of 0s, 1s and 2s :

~ Given an array of size N containing only 0s, 1s and 2s, sort the array in ascending order using optimized time complexity.

=> Detection Of Cycle in Undirected Graph :

~ Given an undirected graph with V vertices and E edges, check whether it contains any cycle or not.

=> Detect Cycle in Directed Graph :

~ Given a directed graph, check whether the graph contains at least one cycle, else return False

=> Binary Tree Level Order Traversal :

~ Given the root of a binary tree, return the level order traversal of its nodes' values.

=> Binary Tree Zigzag Level Order Traversal :

~ Given the root of a binary tree, return the zigzag level order traversal of its node values.

=> Intersection Of Two Linked Lists :

~ Given the head of two singly linked lists, return the node at which the two lists intersect. If the two linked lists have no intersection at all, return null

=> Rotate Image :

~ Given an n -by- n matrix representing an image, rotate the image by 90 degrees(clockwise).

=> Smallest Number Of Perfect Squares :

~ Given positive integers n , find the smallest number of perfect squares that sum up to n . For example if $n = 13$, you should return 2, since $13 = 9 + 4$

=> Application of Graph :

~ Demonstrate any real-world application of graph?

=> Anagrams :

~ Given an array of strings, return all groups of strings that are anagrams.

=> Leaf at the same level :

~ Given a Binary Tree, check if all the leaves are at the same level or not.

=> Reverse Words in a String :

~ Given an input string s , reverse the order of the words.

=> Spiral Matrix :

~ Given an m -by- n matrix, return all the elements of the matrix in spiral order.

=> Sorted Input array :

~ Given an array of integers that is already sorted in non-decreasing order, find two numbers such that they add up to a specific target number. Let these two numbers be $numbers[index_1]$ and $numbers[index_2]$ where $1 \leq first < second \leq numbers.length$

=> Array vs Linked List :

~ How is an Array different from Linked List?

=> Lowest Common Ancestor :

~ Given a Binary Tree, find the lowest common ancestor in a binary tree

=> Next Greater Element :

~ Given an array, find the next greater element for every element. For example $[3, 4, 1, 10]$, the next greater element for each element $[4, 10, 10, -1]$.

=> Edit Distance :

~ Given two strings $str1$ and $str2$, find the minimum number of edits required to convert $str1$ into $str2$. For example $str1 = "prya"$ and $str2 = "Priya"$, we can convert $str1$ to $str2$ by inserting 'i'.

=> Binary Tree is BST or not :

~ Given a Binary Tree, check if a Binary Tree is BST or not.

=> Maximum number of nodes :

~ What is the maximum number of nodes in a binary tree of height k ?

=> Single Number :

~ Given an array of integers, every element appears twice except for one. Find that single element in an array. For example $arr = [1, 3, 3, 2, 2]$, output = 1

=> Linked List :

~ What is the primary advantage of Linked List?

=> Height Of Binary Tree :

~ Given a Binary Tree, calculate the height of the Binary Tree

=> Remove Duplicates from Sorted Array :

~ Given an array of integers sorted in non-decreasing order, remove the duplicates in place such that unique elements appear only once, and the relative order should be kept the same.

=> Build Heap :

~ What is the time complexity of building a heap, and also, Give a mathematical intuition behind the time complexity of Build Heap?

=> Merge Sorted Array :

~ Given two integer arrays, $arr1$ and $arr2$, sorted in non-decreasing order, you have to merge these two arrays into one sorted array in non-decreasing order.

=> Heap :

~ What is the advantage of the heap over a stack?

=> Sorting :

~ What is the meaning of a stable and unstable sorting algorithm? Demonstrate it with the help of some sorting algorithm.

=> Matrix Multiplication :

~ Can we do the task of matrix multiplication in less than $O(n^3)$ time complexity? If yes, then how and what's the optimized time complexity we can get from that?

=> Inplace vs Outplace Sorting Algorithm :

~ Can you explain the difference between in place and outplace sorting algorithms, and is there any algorithm you know so far which is outplace sorting algorithm?

=> Non-Comparison Sorting Algorithm :

~ Can you explain what a non-comparison-based sorting algorithm is with an example?

=> Tree vs Graph :

~ What is the difference between Tree and Graph-based Data Structure?

=> Topological Sort :

~ Can you explain the topological sort in a graph and where it is used practically?

=> Longest Common Prefix :

~ Write a function to find the longest common prefix string amongst an array of strings. For example, $str = [Priya, Priyanka, Priyanshu]$, then the output should be "Priya".

=> Palindrome in Linked List :

~ Given the head of a singly linked list, return true if it is a palindrome.

=> Minimum Value in Stack :

~ Design a Stack Data Structure that supports Push, Pop, Top and give us the minimum value in a stack in constant time.

=> Longest Palindromic Substring :

~ Given a string s , return the longest palindromic substring in s . For example : $str = "babad"$, so output is "bab".

=> Kth Smallest Value in Binary Search Tree :

~ Given the root of BST and an integer k , return the k th smallest value

=> Divide two integers :

~ Given two integer dividends and divisor, divide two integers without using multiplication, division and mod operator.

=> Positive Missing Integer :

~ Given an unsorted integer array, return the smallest positive integer. For example : $arr = [1, 2, 0]$ output is 3.

=> Stable vs Unstable Sorting Algorithm :

~ If the user's requirement is a minimum number of swaps, then which sorting algorithm should we prefer?

=> Binary Search :

~ Can we implement Binary Search in Linked List? If yes, then how and If not, then is there any alternative of Binary Search in Linked List which gives

almost the same time complexity like Binary Search provides us.

=> Complete vs Almost Complete Binary Tree :

~ *What is the difference between Complete vs Almost Complete Binary Tree?*

=> Heap Data Structure :

~ *When we should go for Minheap or Maxheap based Data Structure.*

=> Huffman Coding :

~ *What is the application of Huffman Coding?*

=> Simple vs Multigraph :

~ *What is the difference between Simple Graph and Multi Graph? Which type of graph is used practically.*

=> Generate Parentheses :

~ *Given n pair of parenthesis, write a function to generate all well-formed combinations of parentheses.*

=> Factorial Trailing Zeros :

~ *Given an integer n , return the number of trailing zeros in $n!$*

=> Spanning Tree in Complete Graph :

~ *Given n , which indicates the number of vertices in a graph, how we can get the number of Spanning Tree in Complete Graph.*

=> Graph Connected or Not :

~ *You are given a graph. How can you determine whether the graph is connected or not?*

=> DFT vs BFT :

~ *What Data Structure is internally used for the implementation of DFT and BFT?*

=> MergeSort :

~ *In MergeSort, if the input is "A" sorted subarrays of each size "B", Then what is the time complexity of a single sorted array.*

=> Cycle in Graph :

~ *Given a graph, check whether it contains a cycle or not?*

=> Sum Greater than 1000 :

~ *Given a sorted array of n elements, find any two elements such that the sum of an element "A" and "B" is greater than 1000.*

=> Maximum Number of Vowels in a Substring :

~ *Given a string s and an integer k , return the maximum number of vowel letters in any substring s with length k .*

=> Complete Binary Tree :

~ *Which data structure is preferable to store the complete binary tree and why?*

=> Minimum nodes in Binary Tree :

~ *What is the minimum number of nodes that a binary tree can have?*

=> Doubly Linked List :

~ *Illustrate any real-life application of Doubly Linked List end to end.*

=> Analysis in Algorithm :

~ *Why do we need to do an algorithm analysis?*

=> AVL Tree vs BST :

~ *How can AVL Tree be useful in various operations as compared to BST?*

=> B-Tree :

~ *Where most of the time B-Tree based data structure is used frequently and how?*

=> Interpolation Search :

~ *Usually, we studied Linear and Binary Search. Do you have an idea about interpolation search and how it's working?*

=> Divide and Conquer vs Dynamic Programming :

~ *Can you explain the major differences that you have observed between Divide and Conquer vs Dynamic Programming, and how do we decide when we should go for which approach?*

=> Recursion :

~ *Which Data Structure is used internally to perform recursion operations?*

=> Hashing :

~ *What is the worst-case time complexity of searching an element in Hash Table?*

=> Postfix Form :

~ *What is the postfix form of $(A + B) * (C - D)$*

=> Tree Data Structure :

~ *What is the real-life applications of Tree-Based Data Structure?*

=> Overlapping Of Two Rectangles :

~ *How to find if two given rectangles overlap or not?*

=> First Non-Repeating Character :

~ *Given a string, find its first non-repeating character. For example : $str = ["mmadlsals"]$, output should be "d".*

Power BI Projects

Topic Name : DATA ANALYTICS

Sub-topic Name : POWER BI PROJECTS

Course link : <https://ineuron.ai/course/Power-BI-Projects>

Course Description :-

This course aims to make you aware of the project on a real scenario basis. Solve real-world business challenges using reports, visualizations, and other analytics tools that allow you to gather and exchange data from diverse domains.

Course Features :-

- => Roadmap
- => Quizzes
- => Assignment
- => Downloadable resources
- => Completion certificate

What you will learn :-

- => Gain knowledge from end-to-end project reports in PowerBI
- => Industry-level experience
- => Power pivots, slicers, power view, data analysis expressions

Requirements :-

- => Prior knowledge of PowerBI tool
- => A system with internet connection
- => Microsoft power BI desktop account
- => Dedication

Instructors :-

=> Khushali Shah :

~ A data scientist having rich experience working with MNCs and start-ups in the field of data science and machine learning. She has expertise in Chatbot development for various domains & been developing professionally for 6+ years with diverse job history. She also had positions in software module development, web app development, functional designs, requirement gathering, client interaction, and server setup/admin & can help everywhere in the stack; she loves wearing multiple hats to an extent. She also believes in enhancing her skills by training and learning new things day by day.

Curriculum details :-

=> Introduction :

~ Overview Preview

=> Spend :

~ Project overview Preview
~ Data load
~ Table report
~ Matrix report
~ Funnel chart
~ Pie chart
~ Scatter plot
~ QnA

=> Product :

~ Project overview
~ Load data
~ Create measure
~ Card total units sold
~ Donut chart
~ World flag

=> Acquisition :

~ Project overview
~ Load data
~ Create measure
~ Card
~ Top performer
~ Region sales
~ Sales last year
~ Profit comparison
~ Moving average

Robotics

Topic Name : K12

Sub-topic Name : CLASS10

Course link : <https://ineuron.ai/course/Robotics>

Course Description :-

The Introduction to Robotics Specialization covers the fundamentals of robot flight and movement, as well as how robots detect their surroundings and alter their movements to avoid obstacles, negotiate tough terrains, and complete complicated jobs like construction and disaster response. You'll learn about real-world examples of how robots have been used in disasters, how they've improved human health care, and what their future capabilities will be. The courses lead up to a capstone in which you'll learn how to programme a robot to execute various tasks like flying and gripping objects.

Course Features :-

- => Live instructor led classess
- => Completion certificate
- => Integrate academic knowledge with the tech
- => Real-time Project
- => Live Class Recording
- => Doubt Clearing
- => Assignment in all the Module
- => Quiz in every Module
- => Career Counselling
- => Completion Certificate

What you will learn :-

- => Introduction
- => Essential Tools Basics
- => Robot Car
- => Introduction to Bluetooth
- => Project: Bluetooth Robot Car

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Curriculum details :-

- => Introduction :
 - ~ *What is Robotics?*
 - ~ *Components Required*
- => Essential Tools Basics :
 - ~ *What is a breadboard?*
 - ~ *Using a Breadboard*
 - ~ *Using a Multimeter*
 - ~ *Using jumper wires*
 - ~ *Soldering Basics*
 - ~ *Servo Motor*
 - ~ *Dc motor*
 - ~ *Stepper Motor*
- => Robot Car :
 - ~ *Components of Robot car*
 - ~ *Assembly of the car*
 - ~ *Coding*
 - ~ *Testing*
- => Introduction to Bluetooth :
 - ~ *Components Using Bluetooth*
 - ~ *Bluetooth module basics*
 - ~ *Bluetooth Coding & Output*
- => Project: Bluetooth Robot Car :
 - ~ *Components of Bluetooth Car*
 - ~ *Assembly*
 - ~ *Coding*

Digital Marketing

Topic Name : DIGITAL MARKETING

Sub-topic Name : DIGITAL MARKETING MASTERS

Course link : <https://ineuron.ai/course/Digital-Marketing>

Course Description :-

Grow your digital marketing results faster through the power of growth of businesses(brand), products, and services. Digital metrics, such as lead analytics, web traffic, Web Analytics, social media marketing, content marketing, Email Marketing, Search Engine marketing, digital conversion of audience ,etc will be covered in this course. You will be able to enhance your skill and you can explore industry-driven practices to keep your self ready for tomorrow.

Course Features :-

- => Learning of different Digital Marketing tools
- => Website development through different website builder tools
- => Content creations, business post and display ads for different marketing platform
- => Technical aspects of SEO, ORM, website audits and report generation
- => Live campaign tracking and analytics on different marketing platform
- => Understanding advertising on well-known social media platforms
- => SMS Marketing Tools and Platforms
- => Promote local business through google local business listing
- => Quiz Questions for knowledge & skill development
- => Activity based Assignments in each Module
- => Completion Certificate

What you will learn :-

- => Basics of Digital Marketing and Business Landing page.
- => Basics of Content Marketing & Content Design
- => Search Engine and Search Engine optimization
- => Search Engine Marketing
- => Web Analytics
- => Social Media Marketing
- => SMS Marketing
- => Email Marketing
- => Local Business Promote and Marketing

Digital Marketing strategy & Plan

Requirements :-

- => No prior knowledge in Digital Marketing
- => A System with Internet Connection
- => Basic Knowledge of HTML
- => Dedication

Instructors :-

=> Ankur Khanna :

~ Highly-motivated, energetic and dynamic Digital Marketing Mentor and Assistant Professor having 7+ years of experience in Digital Marketing Industry. Strong practical knowledge of different digital marketing tools aimed at meeting the needs of diverse groups of learners.

Curriculum details :-

- => Introduction to digital marketing :
 - ~ Understanding definition Preview
 - ~ Why digital marketing
 - ~ Important role in digital marketing
- => Digital marketing history & evolution :
 - ~ History and evolution of digital marketing Preview
- => Traditional marketing intro :
 - ~ Introduction to traditional marketing Preview
- => The audience understanding :
 - ~ Audience in digital marketing

~ Audience in traditional marketing

=> Cost benefit analysis in digital marketing and traditional marketing :

~ Cost analysis in traditional marketing

~ Cost analysis in digital marketing

=> Digital marketing vs traditional marketing :

~ Marketing 4p concept

~ Examples of digital ads

~ Pros and cons

=> Types of digital marketing :

~ Offline or online marketing

~ Marketing platforms

~ Categorization of digital marketing

~ Different type of digital marketing

=> Websites :

~ Website concept

~ Website and webpages concept

~ Static and dynamic website

=> Domain and hosting :

~ What is domain name system

~ Web hosting services

=> Intro to business landing pages :

~ Concept of business landing page

~ Website vs landing page

=> Website development tools :

~ Concept of website development tools

~ Website development process

~ Concept of CMS

~ Concept of website builder tools

=> Website building with godaddy website builder tool :

~ Basic parts of website

~ Approach to design website

~ Basic controls of google site

~ How to create header menu of website

~ How to design header image of website

~ New arrival control in website

~ Divider / section control in website

~ Image slider control in website

~ Youtube video control in website

~ How to create footer in website

~ How to create footer quick links in website

~ Contact us page in website

~ Whatsapp button in website

~ Products page in website

~ Add products in product page

~ Product description page in website

~ Product sections and dividers in website

~ New arrivals page in website

~ Announcement banner control in google sites

~ Website publish control in google site

~ Introduction to godaddy website builder

~ Business website in selection in godaddy website builder tool

~ Basics control in godaddy website builder

~ Control menu understanding

~ Header image / video control

~ Image slider control

~ Featured products section

~ Manage store control

~ Add multiple options in product entry

~ Payment gateway setup

~ How to create featured products category

~ Services setup for a website

~ Blog setup for a website

~ Email subcription and people review section

~ Managing staff for online appointments

~ Backend process for email subcription and image gallery)

~ Understanding coupons

~ Livestream section

~ Video and audio section

~ Pdf viewer control

~ Job posting control

~ Services control

~ Privacy policy control / partners section / event calender

~ Social feed control

~ Checkout and shipping control

~ Tax rules control

~ Publish website control / revision

=> Introduction to content marketing :

~ Module 2 overview

~ Power of content marketing

~ Brand value and awareness

~ What actually content marketing

~ Understanding the defination

- ~ Steps to start content marketing
- ~ Steps to understand
- ~ Live content example

=> Content marketing history :

- ~ History of content marketing

=> Important roles in content marketing :

- ~ Client, content marketer and content marketing company
- ~ Content creators and iphone 12 content example
- ~ Content marketer vs digital marketer

=> Content marketing strategy and its benefits :

- ~ What to publish content
- ~ Strategy and effective strategy
- ~ Steps in strategy
- ~ Reasons for good strategy
- ~ Effective content marketing strategy
- ~ Customer journey stage
- ~ Benefits of content marketing
- ~ Concept of lead generation
- ~ Lead generation process
- ~ Definition and meaning of lead generation
- ~ Online reputation management
- ~ Disadvantage of content marketing

=> Understanding the audience in content marketing :

- ~ Channel and audience
- ~ Distribution plan in audience
- ~ Different channel audience

=> Content marketing relation with traditional marketing :

- ~ Why both are actually relate
- ~ Traditional marketer approach
- ~ Inbound marketing concept
- ~ Outbound marketing
- ~ Customer journey in inbound marketing
- ~ Customer journey stage
- ~ Best contain in customer journey
- ~ Consideration in customer journey
- ~ User generated content in customer journey
- ~ Content marketing vs. traditional marketing

=> Types of content & content marketing channel :

- ~ Different type of content
- ~ Star bucks india case study in content marketing
- ~ Mc donalds case study in content marketing
- ~ Different type of written content
- ~ Different type of video content
- ~ Different type of graphic content
- ~ Wireframes and site mock-ups
- ~ Different type of audio content
- ~ Different type of social content
- ~ Introduction to content marketing channel
- ~ Different content marketing channel
- ~ Different social channels
- ~ Display ads and link building

=> Content marketing distribution strategy :

- ~ Introduction to content distribution strategy
- ~ Content distribution channel
- ~ Owned channels
- ~ Organic and paid channels

=> Introduction to content creation :

- ~ Introduction to content creation
- ~ ROI on different type of content
- ~ Common basis for content creation

=> Tool setup and content designing :

- ~ Introduction to tools setup for content designing
- ~ Different tools for written content
- ~ Hemming way tool for written content
- ~ Grammarly tool for written content
- ~ Content correction and creation in grammarly tool
- ~ Answer the public tool for topics
- ~ Headline analyzer for written content
- ~ Buzzsumo tool for written content
- ~ Google trends tool use in written content
- ~ Keyword planning for written content
- ~ Google keyword planner use in written content
- ~ Different graphic content creation tools
- ~ Snappa tool for graphic content creation
- ~ Instagram post creation using snappa tool
- ~ Heading in instagram post creation using snappa tool
- ~ Different control in snappa tool
- ~ Instagram story creation using snappa tool
- ~ Canva tool for graphic content creation
- ~ Instagram post creation using canva tool

DART

Topic Name : MOBILE DEVELOPEMENT

Sub-topic Name : DART

Course link : <https://ineuron.ai/course/DART>

Course Description :-

Dart is a programming language designed for client development, such as for the web and mobile apps. It is developed by Google and can also be used to build server and desktop applications. Dart is an object-oriented, class-based, garbage-collected language with C-style syntax.

Course Features :-

- => Be able to program in Dart professionally
- => Master the Dart programming language by learning every concepts
- => Be able to build fully fledged apps with flutter(using dart)
- => Learn to use modern frameworks like Flutter in future

What you will learn :-

- => Learn Dart from scratch to Classes & Objects
- => A step towards build mobile apps
- => Learn how to write control flow statements
- => Learn how to compile and debug the code
- => Learn all the basics without stopping after then: Dive deeply into Flutter & Dart and become an advanced developer

Requirements :-

- => No prior experience in anything required.

Instructors :-

- => Syed Ashraf :
 - ~ Full Stack Engineer at TensorGo Technologies

Curriculum details :-

- => Introduction & Starting it up :

- ~ Introduction Preview
- ~ Installation
- ~ Setting it up
- ~ Comments
- ~ Programming Flow

- => Data Types :

- ~ Variables Preview
- ~ String Basics
- ~ Operators
- ~ Var Data Type

- => Collections :

- ~ Lists
- ~ Sets
- ~ Maps

- => Control Flow Statements :

- ~ If-Else
- ~ Loops
- ~ Other Loops
- ~ Break & Continue
- ~ Switch Case

- => Objects :

- ~ Functions
- ~ Classes & Objects

- => Extra :

- ~ Error Handling
- ~ Packages

Tableau Job Preparation

Topic Name : DATA ANALYTICS

Sub-topic Name : TABLEAU INTERVIEW QUESTIONS

Course link : <https://ineuron.ai/course/Tableau-Job-Preparation>

Course Description :-

Tableau enables critical decision-makers to learn to display data and uncover data patterns such as customer purchase behaviour, sales trends, or production bottlenecks. We are presenting an end-to-end education for you to make a difference and join the industry. This course will cover all of the capabilities of Tableau that enable access to explore, experiment with, prepare, and present data quickly and beautifully throughout an organisation.

Course Features :-

=> N

=> u

=> I

=> I

What you will learn :-

=> N

=> u

=> I

=> I

Requirements :-

=> N

=> u

=> I

=> I

Instructors :-

=> Khushali Shah :

~ A data scientist having rich experience working with MNCs and start-ups in the field of data science and machine learning. She has expertise in Chatbot development for various domains & been developing professionally for 6+ years with diverse job history. She also had positions in software module development, web app development, functional designs, requirement gathering, client interaction, and server setup/admin & can help everywhere in the stack; she loves wearing multiple hats to an extent. She also believes in enhancing her skills by training and learning new things day by day.

Curriculum details :-

=> Tableau :

- ~ What is Tableau ? Preview
- ~ What are measures and dimensions ? Preview
- ~ What are the Different tableau Products ?
- ~ What are the Datatypes supported in Tableau ?
- ~ Define LOD Expression.
- ~ What is meant by "discrete" and "continuous" in Tableau ?
- ~ What is the difference between .twb and .twbx extension ?
- ~ What are the different Connections you can make with your dataset ?
- ~ What are filters in Tableau ?
- ~ Compare blending and joining in Tableau.
- ~ Can you explain the Tableau design flow ?
- ~ How is the Context Filter different from other Filters ?
- ~ What are Parameters ?
- ~ Give an Example of a story in Tableau.
- ~ What is aggregation and disaggregation of data ?
- ~ What is Calculated field and how you will create one ?
- ~ What is the difference between a Heatmap and Treemap ?
- ~ What is the use of user roles in Tableau ?
- ~ How do you generally perform load testing in Tableau ?
- ~ What is Tableau data engine ?
- ~ What are the components in a dashboard ?
- ~ How do you embed views into webpages ?
- ~ What is the difference between published data sources and embedded data sources in Tableau?
- ~ How to view underlying data sources in Tableau ?
- ~ State some ways to improve the performance in Tableau ?
- ~ Define dual-axis.
- ~ Define Gantt chart.
- ~ State some reasons for the low performance in Tableau.Explain in details.
- ~ State a few charts that we should not use with valid reasons.
- ~ Do we have any data limitations in Tableau public ?
- ~ Is it possible to view SQL commands generated by Tableau ?

- ~ What are the difference between RANK and INDEX ?
- ~ Can R and Tableau be used together ?
- ~ Can you get values from two different sources as a single input into parameter ?
- ~ Suppose without any line/bar chart I want to design a view to show the region-wise profit and sale. How should I approach it? Explain.
- ~ How to generate longitude and latitude in Tableau ?
- ~ How to increase size of pie in Tableau ?
- ~ Why use a hierarchical field in Tableau ?
- ~ List the types of maps available in Tableau.

C++ Bootcamp for Beginners

Topic Name : PROGRAMMING

Sub-topic Name : C++

Course link : <https://ineuron.ai/course/C++-Bootcamp-for-Beginners>

Course Description :-

Ultimate modern C++ Bootcamp. A modern approach to understand C++. By mastering the fundamentals of the language, you can start writing C++ programmes right away. Additionally, you will hear several methods and viewpoints on using C++ professionally.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => Introduction to CPP
- => Getting Started with CPP
- => Basics but indepth of CPP
- => Functions in CPP
- => Object Oriented Programming
- => Smart Pointers in CPP

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Instructors :-

=> Hitesh Choudhary :

~ I like to make videos related to code and tech in my free time. I also lead a few tech teams in startups, help in hiring talent for companies. I am also on a part time traveller, with 31 countries checked off so far!

Curriculum details :-

=> Introduction to CPP :

- ~ CPP20 A note
- ~ CPP20 section1
- ~ Welcome to Cpp bootcamp
- ~ Prerequisite and tools for cpp
- ~ Understand the entry point of hello world
- ~ Compare the 2 hello world
- ~ Version history and official documentation of cpp

=> Getting Started with CPP :

- ~ CPP20 section2
- ~ Return type and comments
- ~ Redefining program in cpp
- ~ What is namespace in cpp
- ~ First iteration of program
- ~ Can I name that
- ~ Get the color and assignment

=> A Little fast pace CPP :

- ~ CPP20 section3
- ~ Your first introduction to pointers
- ~ Reference is the actual tough thing in cpp
- ~ Cpp array are different with pointers
- ~ A formal introduction to integers
- ~ Conditionals and ternary
- ~ Conditionals as switch
- ~ While and do while loops
- ~ Introduction to for and range based for loops
- ~ Loop with pointers and shortcuts

=> Basics but indepth of CPP :

- ~ CPP20 section4

- ~ Always use float with caution
- ~ Why always divide by zero for try catch block
- ~ Sneek peek to functions in cpp
- ~ linkers qualifiers prefix and postfix
- ~ Basics of operations on cpp
- ~ Logical AND OR and NOT
- ~ bitwise operation in cpp
- ~ Memory leaks in cpp

=> More datatypes in CPP :

- ~ CPP20 section5
- ~ Get started with structs in cpp
- ~ Enums and Preprocessors
- ~ A challenge to strongly types language
- ~ Heap and Stack memory with a version discussion

=> Functions in CPP :

- ~ CPP20 section6
- ~ Detailed introduction to functions
- ~ How to create a header file in cpp
- ~ Your first introduction to templates
- ~ What are functional pointers
- ~ nullptr saves the day
- ~ Factorial and recursion are close friend
- ~ Lets talk about MACROS
- ~ Variadic templates and recursion

=> Object Oriented Programming :

- ~ CPP20 section7
- ~ A design example
- ~ Get started with class and objects
- ~ Getters and Setters for a data member
- ~ Method separation and const qualified methods
- ~ Constructor destructor and rule of 3
- ~ Disable the constructor
- ~ THIS is not easy in cpp

=> Little more OOPS :

- ~ CPP20 section8
- ~ Inheritance is my favourite
- ~ Base class Derived class and overriding
- ~ Friend keyword come with caution
- ~ Multiple Inheritance
- ~ polymorphism and virtual

=> Smart Pointers in CPP :

- ~ CPP20 section9
- ~ What are smart pointers
- ~ Unique pointers and issues
- ~ Shared pointers in smart pointers
- ~ Weak pointers in smart pointers

=> Move Semantics file & lambda :

- ~ CPP20 section10
- ~ Move semantics Lvalue and Rvalue
- ~ Vectors - Dynamic array from STD template library
- ~ Lambda - a small hello
- ~ Create, rename and delete files
- ~ Reading and writing into files and MODES

=> STL - Standard Template Library :

- ~ CPP20 section11
- ~ Introduction to STL and generic programming
- ~ Main components in STL
- ~ Functors in STL
- ~ SORT algorithms in STL
- ~ SEARCH algorithms in STL
- ~ Partition and Stable partition in STL

=> STL - a little more :

- ~ CPP20 section12
- ~ Revisiting vectors in STL
- ~ List in STL
- ~ Queue and priority queue in STL
- ~ Deque in STL
- ~ Stack in STL and assignment
- ~ Sets and MultiSets in STL
- ~ MAPS and assignment

Data Structure and Algorithm Interview Preparation

Topic Name : DATA STRUCTURE

Sub-topic Name : DSA INTERVIEW

Course link : <https://ineuron.ai/course/Data-Structure-and-Algorithm-Interview-Preparation>

Course Description :-

This course is designed mostly for Data structure and Algorithms test takers.

Course Features :-

=> Quizzes

=> Course completion certificate

What you will learn :-

=> DSA Theoretical Test

=> DSA Practical Test

=> DSA Aptitude Test

Requirements :-

=> System with minimum i3 processor or better

=> At least 4 GB of RAM

=> Working internet connection

=> Dedication to solve

Curriculum details :-

=> Data structure and Algorithms Test :

- ~ DSA Test 1
- ~ DSA Test 2
- ~ DSA Test 3
- ~ DSA Test 4
- ~ DSA Test 5
- ~ DSA Test 6
- ~ DSA Test 7
- ~ DSA Test 8
- ~ DSA Test 9
- ~ DSA Test 10
- ~ DSA Test 11
- ~ DSA Test 12
- ~ DSA Test 13
- ~ DSA Test 14
- ~ DSA Test 15
- ~ DSA Test 16
- ~ DSA Test 17
- ~ DSA Test 18
- ~ DSA Test 19
- ~ DSA Test 20

Data Science Interview

Topic Name : DATA SCIENCE

Sub-topic Name : MACHINE LEARNING INTERVIEW

Course link : <https://ineuron.ai/course/Data-Science-Interview>

Course Description :-

This course is designed for an individual trying to transition towards various data science careers in the industry. Keeping all the hurdles in mind that we generally face during your transition so that your journey will be smooth and without losing any opportunity, you will be able to transition in the industry. Discuss, Collaborate, Participate and Win the Race.

Course Features :-

- => Online Instructor-led learning
- => Meet with Achiever
- => Proper Roadmap
- => One-One Resume Building
- => Lifetime Dashboard access
- => Doubt clearing
- => Quiz in every module
- => Career Counselling
- => Assessments
- => Mock Interview
- => Certificate
- => 850 + interview question live discussion
- => AI leader talk(Panasonic, EY, Verizon, Apple, and many)

What you will learn :-

- => Profile Building
- => System Designing
- => Domain Understanding
- => Common Mistakes
- => Project Management
- => ML Interview Questions
- => DL Interview Questions
- => NLP Interview Questions
- => Stats Interview Questions
- => Python Interview Questions
- => Computer Vision Interview Questions
- => Mock Interview

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication
- => Basic Understanding of Python
- => Basic Understanding of ML
- => Basic Understanding of DL

Instructors :-

=> krish naik :

~ Having 10+ years of experience in Data Science and Analytics with product architecture design and delivery. Worked in various product and service based Company. Having an experience of 5+ years in educating people and helping them to make a career transition.

=> Sudhanshu Kumar :

~ Having 8+ years of experience in Big data, Data Science and Analytics with product architecture design and delivery. Worked in various product and service based Company. Having an experience of 5+ years in educating people and helping them to make a career transition.

Curriculum details :-

=> Introduction about Data science industry and does and don't in your profile and public profile building with iNeuron team. :

- ~ *Induction & Course Introduction*
- ~ *Impact of Data Science in today's world & Roles in Data Science*

=> Python Interview Questions :

- ~ *50 Interview Questions Day1*
- ~ *50 Interview Questions Day2*

=> Stats Interview Questions :

- ~ *60 Interview Questions Day1*
- ~ *60 Interview Questions Day2*

=> Machine Learning Interview Question and Solution Design :

- ~ *40 Interview Questions Day1*
- ~ *40 Interview Questions Day2*
- ~ *40 Interview Questions Day3*
- ~ *40 Interview Questions Day4*
- ~ *40 Interview Questions Day5*
- ~ *40 Interview Questions Day6*
- ~ *40 Interview Questions Day7*

=> Deep Learning & Computer Vision Interview Questions :

- ~ *40 Interview Questions Day1*
- ~ *40 Interview Questions Day2*
- ~ *40 Interview Questions Day3*
- ~ *40 Interview Questions Day4*
- ~ *40 Interview Questions Day5*

=> NLP Interview Questions :

- ~ *40 Interview Questions Day1*
- ~ *40 Interview Questions Day2*

=> Project deployment & Solution Design Life Cycle Interview Questions :

- ~ *50 Interview Questions Day1*
- ~ *50 Interview Questions Day2*

=> Meet with multiple people who has made recent transition :

- ~ *Ask anything*
- ~ *Get suggestion and roadmap*

=> Generic Project architecture design for interview :

- ~ *How Project start in Industry?*
- ~ *Business Expectation*
- ~ *Data Sharing Agreement*
- ~ *Proof of Concept*
- ~ *Master Data Management*
- ~ *High Level Architecture Design*
- ~ *Low Level Architecture Design*
- ~ *Project Wireframe*
- ~ *Data Accusation*
- ~ *Code Level Architecture*
- ~ *Tech Identification*
- ~ *Team Building*
- ~ *Project Delivery Methodology*
- ~ *Project Timeline Calculation*
- ~ *Infrastructure Setup*
- ~ *Project Cost Estimation*
- ~ *Project Kickoff*

=> Resume Design and projects by iNeuron one to one resume building :

- ~ *Resume Template Selection*
- ~ *Tech Stack Involvement*
- ~ *Project Selection & Alignment as per your Experience*
- ~ *Project Details*
- ~ *Your Involvement in Project*
- ~ *Tech Stack for Project*
- ~ *Fine Tuning of your Resume*
- ~ *Proof Reading*
- ~ *LinkedIn & GitHub Update*
- ~ *Applying for Job*
- ~ *Resume Finalization based on Job Description*
- ~ *One to One Discussion with iNeuron Team*

=> Mock interview with Krish and Sudhanshu one to one live/Offline :

- ~ *Fact Check*
- ~ *All round Interview*
- ~ *Review*
- ~ *Feedback*
- ~ *Suggestions*

=> Interaction with many achievers who has done a recent transition in data science on all level :

- ~ *Interaction with Ineuron Achievers of all Ages*
- ~ *Ask Anything*
- ~ *Expert Advice*
- ~ *Doubt Clarification*

=> Final touch of everything for next journey and launch :

- ~ *Check the JOSH!!!*

Topic Name : CLOUD

Sub-topic Name : AWS

Course link : <https://ineuron.ai/course/AWS>

Course Description :-

Welcome to AWS Cloud Practitioner Essentials. If you're new to the cloud, whether you're in a technical or non-technical role such as finance, legal, sales, and marketing, this course will provide an understanding of fundamental AWS Cloud concepts to help you gain confidence to contribute to your organization's cloud initiatives. This course is also the starting point to prepare for your AWS Certified Cloud Practitioner certification whenever it's convenient for you.

Course Features :-

- => Detailed description of all the topics
- => Quizzes
- => Source code
- => Completion certificate

What you will learn :-

- => Summarize the working definition of AWS
- => Differentiate between on-premises, hybrid-cloud, and all-in cloud models
- => Describe the basic global infrastructure of the AWS cloud
- => Explain the six main benefits of the AWS cloud
- => Describe and provide an example of the core AWS services, including compute, network, database, and storage services
- => Identify an appropriate solution using AWS cloud services for various use cases.
- => Describe the AWS well-architected framework
- => Explain the AWS shared responsibility model
- => Describe the core security services within the AWS cloud
- => Describe the basics of AWS cloud migration
- => Articulate the financial benefits of the AWS cloud for your organization's cost management
- => Define the core billing, account management, and pricing models
- => Explain how to use pricing tools to make cost-effective choices for AWS services

Requirements :-

- => No prior knowledge in AWS
- => A system with internet connection
- => Your dedication
- => Interest to learn

Instructors :-

- => MD Imran :
 - ~ Working as Data Scientist with experience in solving real world business problems across different domains.

Curriculum details :-

- => Introduction :
 - ~ Course introduction Preview
 - ~ Creating an AWS account
- => What is cloud computing :
 - ~ The different types of cloud computing Preview
 - ~ AWS cloud overview
 - ~ Console and services in AWS
 - ~ Shared responsibility model and AWS acceptable policy
- => Identity and access management :
 - ~ IAM Introduction: Users, Groups, Policies Preview
 - ~ IAM Users & Groups hands-on
 - ~ IAM Policies
 - ~ IAM Policies hands-on
 - ~ IAM MFA overview
 - ~ IAM MFA hands-on
 - ~ AWS Access Keys, CLI and SDK
 - ~ AWS CLI setup on Windows
 - ~ AWS CLI setup on Mac
 - ~ AWS CLI setup on Linux

- ~ AWS CLI hands-on
- ~ AWS CloudShell: Region Availability
- ~ AWS CloudShell
- ~ IAM Roles for AWS services
- ~ IAM Roles hands-on
- ~ IAM security tools
- ~ IAM security tools hands-on
- ~ IAM best practices
- ~ Shared responsibility model for IAM
- ~ IAM summary

=> EC2 Elastic compute cloud :

- ~ AWS Budget setup
- ~ EC2 basics
- ~ EC2 instance types basics
- ~ Security groups & classic ports overview
- ~ Security groups hands-on
- ~ SSH overview
- ~ How to SSH using Windows
- ~ How to SSH using Windows 10
- ~ EC2 instance connect
- ~ EC2 instance roles demo
- ~ EC2 instance launch types
- ~ Shared responsibility model for EC2
- ~ EC2 summary

=> EC2 instance storage :

- ~ EBS overview
- ~ EBS hands-on
- ~ EBS snapshots overview
- ~ EBS snapshots hands-on
- ~ AMI overview
- ~ AMI hands-on
- ~ EC2 image builder overview
- ~ EC2 image builder hands-on
- ~ EC2 instance store
- ~ EFS overview
- ~ Shared responsibility model for EC2 storage
- ~ Amazon FSx overview
- ~ EC2 instance storage summary

=> Elastic load balancing and auto-scaling groups :

- ~ High availability, scalability, elasticity
- ~ Elastic load balancing (ELB) overview
- ~ Application load balancer (ALB) hands-on
- ~ Application load balancer (ALB) hands-on part 2 error handling
- ~ Auto scaling groups (ASG) overview
- ~ Auto scaling groups (ASG) hands-on
- ~ Auto scaling groups (ASG) strategies
- ~ Section cleanup
- ~ ELB & ASG summary

=> S3 :

- ~ S3 overview
- ~ S3 hands-on
- ~ S3 security: bucket policy
- ~ S3 security: bucket policy hands-on
- ~ S3 website overview
- ~ S3 website hands-on
- ~ S3 versioning overview
- ~ S3 versioning hands-on
- ~ S3 server access logging
- ~ S3 server access logging hands-on
- ~ S3 replication overview
- ~ S3 replication hands-on
- ~ S3 storage classes overview
- ~ S3 storage classes hands-on
- ~ S3 Glacier vault lock & S3 object lock
- ~ S3 encryption
- ~ Shared responsibility model for S3
- ~ AWS snow family overview
- ~ AWS snow family hands-on
- ~ Storage gateway overview
- ~ S3 summary

=> Databases and analytics :

- ~ Databases introduction
- ~ RDS & aurora overview
- ~ RDS hands-on
- ~ RDS deployments options
- ~ ElastiCache overview
- ~ DynamoDB overview
- ~ DynamoDB hands-on
- ~ DynamoDB global tables
- ~ Redshift overview
- ~ EMR overview
- ~ Athena overview
- ~ QuickSight overview
- ~ DocumentDb overview
- ~ Neptune overview

- ~ QLDB overview
- ~ Managed blockchain overview
- ~ DMS overview
- ~ Glue overview
- ~ Databases & analytics summary

=> ECS, Lambda, Batch and Lightsail :

- ~ What is Docker?
- ~ ECS, Fargate & ECR overview
- ~ Serverless introduction
- ~ Lambda overview
- ~ Lambda hands-on
- ~ API gateway overview
- ~ Batch overview
- ~ Lightsail overview
- ~ Lightsail hands-on
- ~ Other compute - summary

=> Deployment and managing infrastructure at scale :

- ~ Cloudformation overview
- ~ Cloudformation hands on
- ~ CDK overview
- ~ Beanstalk overview
- ~ Beanstalk hands-on
- ~ CodeDeploy overview
- ~ CodeCommit overview
- ~ CodeBuild overview
- ~ CodePipeline overview
- ~ CodeArtifact overview
- ~ CodeStar overview
- ~ Cloud9 overview
- ~ CodeStar & cloud9 hands-on
- ~ Systems manager (SSM) overview
- ~ SSM session manager
- ~ OpsWorks overview
- ~ Deployment summary

=> Leveraging the AWS global infrastructure :

- ~ Why global applications?
- ~ Route 53 overview
- ~ Route 53 hands-on
- ~ CloudFront overview
- ~ CloudFront hands-on
- ~ S3 transfer acceleration
- ~ AWS global accelerator
- ~ AWS outposts
- ~ AWS wavelength
- ~ AWS local zones
- ~ Global applications architecture
- ~ Leveraging the AWS global infrastructure summary

=> Cloud integrations :

- ~ Cloud integrations overview
- ~ SQS overview
- ~ SQS hands-on
- ~ SNS overview
- ~ SNS hands-on
- ~ Kinesis overview
- ~ Amazon MQ overview
- ~ Cloud integrations summary

=> Cloud monitoring :

- ~ CloudWatch metrics & CloudWatch alarms overview
- ~ CloudWatch metrics & CloudWatch alarms hands-on
- ~ CloudWatch logs overview
- ~ CloudWatch logs hands-on
- ~ CloudWatch events / eventbridge overview
- ~ CloudWatch events / eventbridge hands on
- ~ CloudTrail overview
- ~ CloudTrail hands-on
- ~ X-Ray overview
- ~ CodeGuru overview
- ~ Service health dashboard
- ~ Personal health dashboard
- ~ Cloud monitoring summary

=> VPC and networking :

- ~ VPC overview
- ~ VPC, subnet, internet gateway & NAT gateways
- ~ Security groups & network access control list (NACL)
- ~ VPC flow logs & VPC peering
- ~ VPC endpoints - interface & gateway (S3 & DynamoDb)
- ~ Direct connect & site-to-site VPN
- ~ Transit gateway overview
- ~ VPC & networking summary

=> Security and compliance :

- ~ Shared responsibility model: reminders & examples
- ~ DDoS protection: WAF & shield
- ~ Penetration testing
- ~ Encryption with KMS & CloudHSM

- ~ *Encryption with KMS & CloudHSM hands-on*
- ~ *AWS certificate manager (ACM) overview*
- ~ *Secrets manager overview*
- ~ *Artifact overview*
- ~ *GuardDuty overview*
- ~ *Inspector overview*
- ~ *Config overview*
- ~ *Macie overview*
- ~ *Security hub overview*
- ~ *Amazon detective overview*
- ~ *AWS abuse*
- ~ *Root user privileges*
- ~ *Security & compliance summary*

=> Machine learning :

- ~ *Recognition overview*
- ~ *Transcribe overview*
- ~ *Polly overview*
- ~ *Translate overview*
- ~ *Lex + connect overview*
- ~ *Comprehend overview*
- ~ *SageMaker overview*
- ~ *Forecast overview*
- ~ *Kendra overview*
- ~ *Personalize overview*
- ~ *Machine learning summary*

=> Account management, billing and support :

- ~ *Organizations overview*
- ~ *Organizations hands-on*
- ~ *Organizations consolidated billing*
- ~ *AWS control tower overview*
- ~ *AWS control tower hands-on*
- ~ *Pricing models of the cloud*
- ~ *Savings plan overview*
- ~ *Compute optimizer overview*
- ~ *Billing & costing tools overview*
- ~ *Estimating costs in the cloud - TCO calculator & pricing calculator*
- ~ *Tracking costs in the cloud - billing dashboard, cost allocation tags, reports*
- ~ *Monitoring costs in the cloud - billing alarms & AWS budgets*
- ~ *AWS trusted advisor*
- ~ *Support plans for AWS*
- ~ *Account best practices summary*
- ~ *Billing summary*

=> Advanced identity :

- ~ *Security token service (STS) overview*
- ~ *Cognito overview*
- ~ *Directory services overview*
- ~ *Single sign-on (SSO) overview*
- ~ *Advanced identity - summary*

=> Other services :

- ~ *Other services - section intro*
- ~ *Workspaces overview*
- ~ *Appstream 2.0 overview*
- ~ *Sumerian overview*
- ~ *IoT core overview*
- ~ *Elastic transcoder overview*
- ~ *Device farm overview*
- ~ *AWS backup overview*
- ~ *Disaster recovery strategies*
- ~ *Cloudendure overview*
- ~ *AWS datasync*

=> AWS architecting and ecosystem :

- ~ *AWS whitepapers well-architected framework*
- ~ *1st pillar: operational excellence*
- ~ *2nd pillar: security*
- ~ *3rd pillar: reliability*
- ~ *4th pillar: performance efficiency*
- ~ *5th pillar: cost optimization*
- ~ *AWS well-architected tool*
- ~ *Right-Sizing*
- ~ *AWS ecosystem*
- ~ *AWS knowledge center*

Backend Development with Django-8 Projects

Topic Name : WEB DEVELOPEMENT

Sub-topic Name : DJANGO

Course link : <https://ineuron.ai/course/Backend-Development-with-Django-8-Projects>

Course Description :-

The most widely used Python web development framework is Django. Django is a Python framework that covers all elements of web development, from handling requests and answers to creating dynamic HTML pages using templates and making database access and maintenance simple. This course has it all baked in, and it's all covered in excellent depth. Django is taught from the ground up in this course. We'll start from the beginning and work our way up, learning how to construct Django projects, execute them, and add functionality step by step.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => Webpages in Django
- => Admin control
- => Sliders
- => Search feilds
- => Navbars
- => Components
- => User authentication
- => Facebook and Google authentication
- => Django message frameworks
- => Django contact forms

Requirements :-

- => System with minimum i3 processor or better
- => At least 4 GB of RAM
- => Working internet connection
- => Dedication to learn

Instructors :-

- => Hitesh Choudhary :

~ I like to make videos related to code and tech in my free time. I also lead a few tech teams in startups, help in hiring talent for companies. I am also on a part time traveller, with 31 countries checked off so far!

Curriculum details :-

- => Getting tools ready for Django :

- ~ Tools that we need
- ~ Your first HelloWorld project in Django
- ~ Understanding file structure

- => Project 1 - Getting basics done :

- ~ our very first Django App
- ~ Configuring new app

- => Project 2 - All about templating basics :

- ~ Project 2 and templating basics
- ~ Rendering form HTML Page
- ~ Adding an about us page
- ~ Adding contact Us page
- ~ Extending pre built templates

- => Project 3 - Interaction with Database :

- ~ Setting up command project

- ~ Creating our first model for sqlite3
- ~ Registering models to admin
- ~ Create read update and delete from database

=> Project 4 - Blog with static pages and Unique URL :

- ~ Articles app creation
- ~ ForeignKey and many to one relation
- ~ Adding articles in database
- ~ Configuring URLs
- ~ Setting up home page
- ~ Adding static files for css
- ~ Unique URL for articles

=> Project 5 - Handling forms and taking input from user :

- ~ Handling user input via forms - setup
- ~ Preparing models and admin
- ~ fixing views and other issues
- ~ Making templates look great with static files
- ~ fetching data from database
- ~ Interaction with Django forms
- ~ Taking input from user and storing it in database

=> Project 6- A CRUD Blog from user input :

- ~ Jump start of CRUD project
- ~ Reusing admin fields in web page
- ~ Update from database
- ~ Delete from database and reverse lazy urls

=> Project 7 - A TODO list with sqlite3 :

- ~ How we will take down this TODO

=> Project 7 - A TODO list with sqlite4 :

- ~ Create models for database

=> Project 7 - A TODO list with sqlite5 :

- ~ Setting up templates and static files

=> Project 7 - A TODO list with sqlite6 :

- ~ Adding a todo and decorators

=> Project 7 - A TODO list with sqlite7 :

- ~ Finishing up Todo

=> Project 8 - Login, Logout and SignUp :

- ~ Setup URL for login
- ~ Having a login View
- ~ A working Login and logout
- ~ A complete signup app

Blockchain Foundations

Topic Name : BLOCKCHAIN

Sub-topic Name : BLOCKCHAIN MASTERS

Course link : <https://ineuron.ai/course/Blockchain-Foundations>

Course Description :-

Presenting the blockchain community session where students will learn the fundamentals of Blockchain Technology along with Solidity programming fundamentals with hands-on practical problems. Learners will learn to build their own cryptocurrency after completion of this community session

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => Introduction to blockchain
- => Ethereum and Solidity
- => Solidity
- => Create your Cryptocurrency Project

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Instructors :-

=> Sanjeevan Thorat :

~ Data Scientist and Blockchain developer, with experience in developing and managing end to end solutions. I have hands-on experience in Python Programming Language, Machine Learning Deep Learning and Natural language processing. Blockchain development experience in smart contracts, Decentralised Finance applications, DAOs, NFTs and Oracles running on Ethereum and Polygon blockchains. I have worked with various clients for different industry projects in the blockchain space. I specialize in building smart contracts on the Ethereum blockchain along with JavaScript integration for enhancing user experience to generate maximum returns on investment.

Curriculum details :-

=> Introduction to blockchain :

- ~ What is Blockchain
- ~ History of Blockchain
- ~ Bitcoin Blockchain in depth

=> Ethereum and Solidity :

- ~ History Ethereum Blockchain
- ~ Ethereum Blockchain in depth
- ~ Creating a blockchain from scratch in Javascript

=> Solidity :

- ~ What is Solidity
- ~ Solidity basics
- ~ Smart contract fundamentals
- ~ Payable functions
- ~ Fallback functions
- ~ View functions
- ~ Pure functions
- ~ Function overloading
- ~ Function overriding
- ~ Solidity Events
- ~ Block and Transaction details
- ~ Solidity Inheritance
- ~ Single Inheritance
- ~ Multiple Inheritance
- ~ Heirarchical Inheritance
- ~ Multilevel Inheritance
- ~ Abstract Contracts
- ~ Solidity Interfaces
- ~ Solidity Libraries

=> Project :

~ *Creating a cryptocurrency with ICO in Solidity from scratch*

Machine Learning in R

Topic Name : DATA SCIENCE

Sub-topic Name : MACHINE LEARNING

Course link : <https://ineuron.ai/course/Machine-Learning-in-R>

Course Description :-

The Machine Learning with R course has been specifically developed to aid in the development of a solid understanding of the fundamentals of machine learning. You'll learn how to prepare data for modeling, train your models, visualize and evaluate their performance, and fine-tune their parameters for improved results. Learn the abilities you'll need to work as a machine learning scientist.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => Rstudio
- => Histograms
- => Scatterplots
- => overlaying plots
- => Simple Linear Regression
- => Multiple Linear Regression
- => Logistic Regression
- => Support Vector Machine
- => Decision Tree Classification

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Instructors :-

- => MD Imran :
 - ~ Working as Data Scientist with experience in solving real world business problems across different domains.

Curriculum details :-

=> Basics of Statistical Computing in R :

- ~ Installing R
- ~ Rstudio
- ~ Packages
- ~ Plot()
- ~ Bar Charts
- ~ Histograms
- ~ Scatterplots
- ~ overlaying plots
- ~ summary()
- ~ describe()
- ~ selecting cases
- ~ data formats part 1
- ~ data formats part 2
- ~ factors
- ~ entering data
- ~ importing data

=> Data Preprocessing in R :

- ~ Getting Started
- ~ Dataset Description
- ~ Importing the Dataset
- ~ Difference between mean and ave
- ~ Taking care of Missing Data part 1

- ~ *Taking care of Missing Data part 2*
- ~ *Encoding Categorical Data*
- ~ *Splitting the dataset into the training set and test set*
- ~ *Feature scaling*

=> Simple Linear Regression :

- ~ *simple Linear Regression Intuition*
- ~ *Simple Linear Regression in R Step 1*
- ~ *Simple Linear Regression in R Step 2*
- ~ *Simple Linear Regression in R Step 3*
- ~ *Simple Linear Regression in R Step 4*

=> Multiple Linear Regression :

- ~ *Multiple Linear Regression in R Step 1*
- ~ *Multiple Linear Regression in R Step 2*
- ~ *Multiple Linear Regression in R Step 3*

=> Logistic Regression :

- ~ *Logistic Regression in R - Step 1*
- ~ *Logistic Regression in R - Step 2*
- ~ *Logistic Regression in R - Step 3*
- ~ *Logistic Regression in R - Step 4*
- ~ *Logistic Regression in R - Step 5*
- ~ *Logistic Regression in R - Step 6*

=> Support Vector Machine :

- ~ *Support Vector Machine Demo*

=> Decision Tree Classification :

- ~ *Decision Tree Classification Demo*

Data Structure and Algorithms with competitive programming

Topic Name : DATA STRUCTURE

Sub-topic Name : DSA MASTERS

Course link : <https://ineuron.ai/course/Data-Structure-and-Algorithms-with-competitive-programming>

Course Description :-

Data Structure and Algorithms for Beginners to Advance entire course will be discussed in python language and all the implementation and project will be done by using python .

Course Features :-

- => Online Instructor-led learning: Live teaching by instructors
- => Every week doubt clearing session after the live classes
- => Lifetime Dashboard access
- => Doubt clearing one to one
- => Assignment in all the module
- => Quiz in every module
- => Everything will be discussed with python

What you will learn :-

- => Analysis in Algorithms
- => Divide and Conquer
- => Greedy Technique
- => Dynamic Programming
- => Arrays
- => Linked List
- => Skip List
- => Hashing
- => Tree
- => Graph Traversal
- => Tree Traversal
- => Programming
- => Stack
- => Queue
- => String Matching
- => NP-Hard and NP-Complete Problems

Requirements :-

- => Dedication
- => PC with internet connectivity

Instructors :-

=> Priya Bhatia :

~ Expertise in data structure competitive programing and solving an analytical problems and implementing data structure algorithm in multiple programing language. I have done my M.Tech in Artificial Intelligence at IIT Hyderabad and have an experience of implementation in multiple projects.

Curriculum details :-

=> Analysis in Algorithms :

- ~ Introduction to Algorithms Preview
- ~ Analyzing Algorithm Preview
- ~ Asymptotic Notation Preview
- ~ Big O
- ~ Omega
- ~ Theta
- ~ Recurrence Relation Solving
- ~ Substitution Method
- ~ Recursive Tree Method
- ~ Master's Theorem

=> Divide and Conquer :

- ~ Introduction to Divide and Conquer
- ~ Discussion of applications of Divide and Conquer
- ~ Finding of maxima and minima

- ~ *Finding Power of an Element*
- ~ *Binary Search*
- ~ *Quicksort*
- ~ *Mergesort*
- ~ *Strassen's Matrix Multiplication*
- ~ *Maximum-subarray problem*
- ~ *Finding of number of inversions*

=> Greedy Technique :

- ~ *Introduction to Greedy Techniques*
- ~ *Discussion of applications of Greedy Technique*
- ~ *Knapsack Problem*
- ~ *Job Sequencing with deadline*
- ~ *Huffman Coding*
- ~ *Optimal Merge Pattern*
- ~ *Minimum Cost Spanning Tree*
- ~ *Kruskal Algorithm*
- ~ *Prim's Algorithm*
- ~ *Single Source Shortest Path*
- ~ *Dijkstra's Algorithm*
- ~ *Bellmanford Algorithm*

=> Dynamic Programming :

- ~ *Introduction to Dynamic Programming*
- ~ *Discussion of applications of Dynamic Programming*
- ~ *Fibonacci Series*
- ~ *Longest Common Subsequence*
- ~ *0/1 Knapsack*
- ~ *Sum of Subset*
- ~ *All Shortest Path*
- ~ *Matrix Chain Multiplication*

=> Arrays :

- ~ *Introduction to Arrays*
- ~ *One Dimensional Array - How to find the address of an element in an array*
- ~ *Two Dimensional Array*
- ~ *Row Major Order*
- ~ *Column Major Order*
- ~ *Searching in an array*
- ~ *Linear Search*
- ~ *Binary Search(Discussed in DAC)*
- ~ *Sorting of an array*
- ~ *Comparison Sort*
- ~ *Selection Sort*
- ~ *Bubble Sort*
- ~ *Insertion Sort*
- ~ *Quicksort(Discussed in DAC)*
- ~ *Mergesort(Discussed in DAC)*
- ~ *Non Comparison Sort*
- ~ *Radix Sort*
- ~ *Bucket Sort*
- ~ *Count Sort*

=> Linked List :

- ~ *Introduction to Linked List*
- ~ *Searching in Linked List*
- ~ *Deleting from a Linked List*
- ~ *Doubly Linked List*
- ~ *Reversal in linked list*

=> Skip List :

- ~ *Introduction to Skip List*
- ~ *Operations and Randomization in Skip Lists*
- ~ *Insertion and Deletion in Skip Lists*
- ~ *Complexity analysis*

=> Hashing :

- ~ *Introduction to Hashing*
- ~ *Hash Tables*
- ~ *Hash Functions*
- ~ *Collision Resolution Techniques*
- ~ *Chaining*
- ~ *Open Addressing*
- ~ *Linear Probing*
- ~ *Quadratic Probing*
- ~ *Double Hashing*
- ~ *Perfect Hashing*
- ~ *Analysis of Chaining*
- ~ *Analysis of Open Addressing*
- ~ *Application of Hashing : Bloom Filters Discussion*

=> Tree :

- ~ *Introduction to Binary Tree*
- ~ *Binary Search Tree*
- ~ *AVL Tree - Creation , Insertion, Deletion*
- ~ *Red Black Tree - Creation , Insertion, Deletion*
- ~ *BTree and B+ Tree - Creation , Insertion, Deletion*

=> Graph Traversal :

- ~ *Breadth First Search*
- ~ *Depth First Search*

=> Tree Traversal :

- ~ Preorder Traversal
- ~ Postorder Traversal
- ~ Inorder Traversal

=> Programming :

- ~ Static and Dynamic Scoping
- ~ Static Variable
- ~ Pointers

=> Stack :

- ~ Introduction to Stack Data Structure
- ~ Implementation of Stack Using Arrays
- ~ Implementation of Stack Using Linked List
- ~ Average Stack Lifetime of an element
- ~ Implementing multiple stacks in single array
- ~ Applications of Stack
- ~ Recursion
- ~ Tail Recursion
- ~ Non-Tail Recursion
- ~ Nested Recursion
- ~ Indirect Recursion
- ~ Infix to Postfix
- ~ Prefix to Postfix
- ~ Postfix Evaluation
- ~ Towers of Hanoi
- ~ Fibonacci Series

=> Queue :

- ~ Introduction to Queue Data Structure
- ~ Implementation of Queue Using Arrays
- ~ Implementation of Queue Using Linked List
- ~ Circular Queue
- ~ Priority Queue
- ~ Implementation of Stack using Queue

=> String Matching :

- ~ Naive String Matching Algorithms
- ~ Rabin-Karp Algorithm
- ~ String Matching with finite automata

=> NP-Hard and NP-Complete Problems :

- ~ NP-Hard
- ~ NP-Complete Problem

System Design with Design Patterns Tech Neuron

Topic Name : SYSTEM DESIGN

Sub-topic Name : SYSTEM DESIGN MASTERS

Course link : <https://ineuron.ai/course/System-Design-with-Design-Patterns-Tech-Neuron>

Course Description :-

The software engineering interview process includes system design questions as a routine element of the process. The way you perform in these interviews reflects on your ability to work with complicated systems, which is reflected in the position and salary offered by the interviewing organisation. The purpose of this course is to help you master software engineering interviews.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => Design principles
- => Introduction and types
- => Creational Patterns
- => Structural Patterns
- => Behavioural Patterns
- => Important System Design Concepts
- => System Design Problems
- => Designing Facebook Messenger
- => Designing Twitter
- => Designing Youtube
- => Designing Netflix

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Curriculum details :-

- => Design principles :
 - ~ *DRY principles*
 - ~ *KISS principles*
 - ~ *SOLID principles*
 - ~ *CUPID principles*
- => Introduction and types :
 - ~ *OOPS overview*
 - ~ *The Singleton Pattern part 1*
 - ~ *The Singleton Pattern part 2*
 - ~ *The Singleton Pattern part 3*
- => Creational Patterns :
 - ~ *The Factory Pattern*
 - ~ *The Factory Method Pattern*
 - ~ *The Abstract Factory Pattern*
 - ~ *The Singleton Pattern*
 - ~ *The Builder Pattern*
 - ~ *The Prototype Pattern*
 - ~ *Summary of Creational Patterns*
- => Structural Patterns :
 - ~ *The Adapter Pattern*
 - ~ *The Bridge Pattern*
 - ~ *The Composite Pattern*
 - ~ *The Decorator Pattern*
 - ~ *The Faade Pattern*

- ~ *The Flyweight Pattern*
- ~ *The Proxy Pattern*
- ~ *Summary of Structural Patterns*

=> Behavioural Patterns :

- ~ *Chain of Responsibility Pattern*
- ~ *The Command Pattern*
- ~ *The Interpreter Pattern*
- ~ *The Iterator Pattern*
- ~ *The Mediator Pattern*
- ~ *The Memento Pattern*
- ~ *The Observer Pattern*
- ~ *The State Pattern*
- ~ *The Strategy Pattern*
- ~ *The Template Pattern*
- ~ *The Visitor Pattern*
- ~ *Null Object pattern*

=> Important System Design Concepts :

- ~ *System Design Basics*
- ~ *Key Characteristics of Distributed Systems*
- ~ *Load Balancing*
- ~ *ClientServer Model*
- ~ *Network Protocols*
- ~ *Storage*
- ~ *Latency And Throughput*
- ~ *Availability*
- ~ *Caching*
- ~ *Data Partitioning*
- ~ *Indexes*
- ~ *Replication*
- ~ *Sharding*
- ~ *Proxies*
- ~ *Redundancy*
- ~ *SQL vs. NoSQL*
- ~ *CAP Theorem and*
- ~ *PACELC Theorem*
- ~ *Consistent Hashing*
- ~ *Long Polling vs WebSockets vs Server Sent Events*
- ~ *Bloom Filters*
- ~ *Quorum, Leader and Follower, Heartbeat, Checksum*
- ~ *Rate Limiting*
- ~ *Logging And Monitoring*
- ~ *Security And HTTPS*
- ~ *API Design*

=> System Design Problems :

- ~ *System Design Interviews: A step by step guide*
- ~ *Designing a URL Shortening service like TinyURL*
- ~ *Designing Pastebin*
- ~ *Designing Instagram*
- ~ *Designing Dropbox*
- ~ *Designing Facebook Messenger*
- ~ *Designing Twitter*
- ~ *Designing Youtube*
- ~ *Designing Netflix*
- ~ *Designing Typeahead Suggestion*
- ~ *Designing an API Rate Limiter*
- ~ *Designing Twitter Search*
- ~ *Designing a Web Crawler*
- ~ *Designing Facebooks Newsfeed*
- ~ *Designing Yelp or Nearby Friends*
- ~ *Designing Uber backend*
- ~ *Designing Ticketmaster*

Publishing Custom Cocoapads

Topic Name : WEB DEVELOPEMENT

Sub-topic Name : FULL STACK WEB DEVELOPMENT

Course link : <https://ineuron.ai/course/Publishing-Custom-Cocoapads>

Course Description :-

This course will help you to publish custom Cocoapads.

Course Features :-

- => Course material
- => Course resources
- => On demand recorded videos
- => Practical exercises
- => Quizzes
- => Assignments
- => Course completion certificate

What you will learn :-

- => How to publish your own cocoapods
- => Installing cocoapods
- => Creating and exploring cocoapods to be published
- => Editing podsec and github push
- => Writing custom function for cocoapods
- => Writing a test case for cocoapods
- => Finally 2C lets push our first cocoapod

Requirements :-

- => System with Internet Connection
- => Interest to learn
- => Dedication

Instructors :-

=> Hitesh Choudhary :

~ I like to make videos related to code and tech in my free time. I also lead a few tech teams in startups, help in hiring talent for companies. I am also on a part time traveller, with 31 countries checked off so far!

Curriculum details :-

=> Cocoapads :

- ~ How to publish your own cocoapods
- ~ Installing cocoapods
- ~ Creating and exploring cocoapods to be published
- ~ Editing podsec and github push
- ~ Writing custom function for cocoapods
- ~ Writing a test case for cocoapods
- ~ Finally 2C lets push our first cocoapod

Build ETL Data Pipeline on AWS EMR Cluster

Topic Name : BIG DATA

Sub-topic Name : BIG DATA PROJECTS

Course link : <https://ineuron.ai/course/Build-ETL-Data-Pipeline-on-AWS-EMR-Cluster>

Course Description :-

With the advent of powerful data warehouses like SnowFlake, BigQuery, redshift spectrum, etc that allow separation of storage and execution, it has become very economical to store data in the data warehouse and then transform them as required. This Project goes over how to design such a ELT system using AWS EMR and Hive. The main objective is to keep the code complexity and server management low, while automating as much as possible

Course Features :-

- => Do Everything In Industry Grade Lab
- => Learn As Per Your Timeline
- => Hands-On Industry Real-Time Projects.
- => Self Paced Learning
- => Dashboard Access

What you will learn :-

- => Real Time Projects
- => Build ETL Data Pipeline on AWS EMR Cluster
- => Components of a Data Engineering Platform
- => Building ETL Pipeline
- => Store data in the data warehouse
- => Build Dashboard using Tableau
- => Hive

Requirements :-

- => System with minimum i3 processor or better
- => At least 4 GB of RAM
- => Working internet connection
- => Dedication to learn

Instructors :-

- => MD Imran :
 - ~ Working as Data Scientist with experience in solving real world business problems across different domains.

Curriculum details :-

- => Welcome to the Course :
 - ~ Course Overview
 - ~ Dashboard Introduction
- => Project :- Build ETL Data Pipeline on AWS EMR Cluster :
 - ~ Introduction of Instructor
 - ~ Introduction to ETL
 - ~ Project Overview
 - ~ End Notes
 - ~ Problem Description
 - ~ Understand the application scope
 - ~ Tour to existing solution
 - ~ End Notes
 - ~ Data Infrastructure: Components used
 - ~ Aws services
 - ~ Data Visualization Tools
 - ~ End Notes
 - ~ Solution Description
 - ~ Data Architecture
 - ~ Tour to Architecture diagram
 - ~ Cost Involved
 - ~ End Notes
 - ~ Exploration of the dataset
 - ~ Creating EMR Cluster
 - ~ Login into EMR hive Project
 - ~ Upload Data into Amazon S3
 - ~ using Hive as ETL Tool
 - ~ Hive Data Insertion
 - ~ CXconnect Tableau to Amazon EMR Hive

- ~ *Plot Charts*
- ~ *Plot Dual Combination Charts*
- ~ *Other Carts*
- ~ *Building Dashboard*
- ~ *End Notes*
- ~ *Conclude the project*
- ~ *Assignments & External Resources*