**Basics (Week 1)**

1. Understanding Internet Concepts
2. How does the internet works?
3. What is HTTP
4. Browsers and how they work?
5. DNS and how it works
6. What is Domian Name
7. What is hosting?

**WebServices (Week 1)**

1. What are web services
2. Different types of Web Services
3. How to Call (Consume) Web Service using plain JavaScript
4. Understand REST Api(https://www.restapitutorial.com/)
   1. http as service transport
   2. various http methods
   3. http status codes (Eg: 200, 403, 404, 500, 503, 504, 502, 405, 505, 301 etc)

**Virtual Machine(Week 1)**

1. What is Virtual Machine?
2. Install Virtual Box (https://www.oracle.com/in/virtualization/technologies/vm/downloads/virtualbox-downloads.html)
3. Install Ubuntu via Virual Box(https://releases.ubuntu.com/18.04/ubuntu-18.04.6-desktop-amd64.iso)
4. Install go lang in Ubuntu VM

**Backend(Go lang)(**Install in Ubuntu VM) - https://golang.org/doc/effective\_go**(Week 2-3)**

1. Go Installation(<https://golang.org/>)
2. go functions
   1. multiple return values
   2. defer
3. go map
4. go error
   1. error
   2. recover
   3. panic
5. go regex
6. go types
   1. struct
   2. interface
   3. pointer
7. go concurrency
   1. concurrency
   2. mutex
   3. channels
   4. atomic variables
8. http server using go
9. json marshalling and unmarshalling using go
10. rest api server in go
11. consuming rest api using go
12. Test Program(go lang) 1: Create a go lang rest server applicaton which should connect to StudentDB(mysql or any DB of choice) for storing and retrieving student details.(Use Postman for testing apis)
    1. POST http://localhost:8080/student

Body – {id: 101, name: student1, class: 5A, marks: 500}

Response: Student Added in Database

* 1. GET <http://localhost:8080/student/101>

Response: {id: 101, name: student1, class: 5A, marks: 500}

* 1. GET <http://localhost:8080/student/>

Response: [{id: 101, name: student1, class: 5A, marks: 500},{id: 102, name: student2, class: 6A, marks: 200}]

* 1. DELETE <http://localhost:8080/student/101>

Response: student deleted from DB

1. Test Program(go lang) 2 Create a go lang http server application
   1. The application should able to read file application.properties with contents -> {port: 9010)
   2. The http server should run in port mentioned in application.properties
   3. When localhost:9010 in browser, response: Hello World

**Docker(Week 4)**

1. Install Docker inside VM
2. What is Docker
3. Docker Architecture
4. What are Docker container, images etc
5. Understand difference between Virtual Machine and docker containers
6. Understand docker in windows and docker in linux
7. DockerFile
8. Understand Docker Features like
   1. Docker build
   2. Docker tagging
   3. Docker run, with different modes like
      1. Detach mode
      2. Interactive mode etc
   4. Docker volume
   5. Exposing ports
   6. Container linking
9. Docker Compose
10. Test Program 1(Docker): Dockerise the second go lang application(Create a docker image). And Docker run the image
    1. Instead of copying application.properties, use docker volume concept to supply application.properties to container
    2. Expose and bind the container port to host port 9055
    3. When localhost:9055 in browser, response: Hello World

**Kubernetes**(https://kubernetes.io/docs/tasks/tools/)**(Week 5)**

1. Install minikube or microk8s inside Ubuntu VM
2. Understand Kubernetes Concepts(<https://kubernetes.io/docs/concepts/>)
3. Test Program 1(Kubernetes): Create a pod using image created in Docker study
   * Use configmap and bind the configmap as volume to supply application.properties to container
   * Use service with nodePort for exposing container port.
   * Use nodePort as 32100
   * When localhost:32100 in browser, response: Hello World

**ReactJS(Week 5 - 6)**

1. What is ReactJS?
2. Understand the difference between some alternatives like Angular, vue.js
3. How React works
4. Understand components
   1. Components and styling
   2. Functional Components and class based components in react
5. Understanding props
6. Handling events like click events
7. Understanding state of compoent
8. Understanding hooks
   1. UseState
   2. useEffect
   3. useContext
   4. useReducer
   5. useRef etc
9. Undestanding routing
   1. Route
   2. Switch
   3. Links and Navigation
10. Consuming rest api
    1. Fetch/Axios -> POST & GET
    2. Parsing Json in javascript
11. Navigating Programatically(useHistory)
12. React Context
    1. Create context
    2. Using context in components
13. Test Program 1(React): Create a simple College management
    1. Able to add student get student using id, delete student using id, list all students
    2. Use golang rest api exposed via kubernetes created in kubernetes study(localhost:32100)