**Day 2**

**Web Service**

Giving the Service for web application when both the application running using different technologies.

Angular or React and backend Express

Amazon : Payment Paytm spring boot

Paypal express js

Gpay Django with python

Phone Pay php

Credit card

Java (request)

App1 XML/JSON App2

JS (Response)

XML : eXtensible markup of language

JSON : JavaScript Object Notation

Types of Web Service

SOAP Web Service : Simple Object Access Protocol : SOAP Web Service base upon SOA (Service Oriented Archiecture).

SB( Service Brocker)

Register (WSDL)

SOAP Req

SP (Service Provider) SC (Service Consumer)

function getAmount(accno) Java

WSDL file SOAP Res

Web Service Description language

.xml

In SOAP base web service consume and produce data only in the form of xml with help SOAP Structure.

Rest Full Web Service : In Rest full Web Service we can consume and produce data in any format base upon client requirement like plain text, html format, json format, xml format or any media type etc.

Rest full web service is not a standard. It expose our resources as a web service.

Node js provided one of the core modules ie http modules.

Express JS : Express JS is a third party module which help to create dynamic web application as well rest full web service using node js.

Create the folder express module using JS

npm init

Or

npm init -y

using node js if we want to connect any RDBMS database node js provided lot of third party modules

for postgres they provided pg module which help to connect post gres database to do operation like insert, delete, update and retrieve using node js program.

Express JS CRUD Operation using typescript

create the folder express-typescript-crud-operation

cd express-typescript-crud-operation

tsc --init tscofig.json file

npm init -y short cut way to create package.json file

npm install express @types/node @types/express

express : this module for js to create express related task.

@types/node : data type with core modules they provide data type of those core modules

@types/express : creating application using typescript in express this in dev mode.

npm install typescript -g

To connect RDBMS database or SQL Database using Node JS

We can use pg for postgres

We can use mysql for mysql database

We can use oracle for oracle database.

We can use Sequelize. It support ORM (Object Relational mapper) like ORM in other programming language hibernate or jpa, nHibernate for .net,

Limitation of pg module. It is native driver. In this module we need to write SQL query. SQL is database dependent.

In programming side object ie JavaScript (model class) and database side table (relation) we are doing mapping or mapper. Using Sequelize module we can do the operation using function rather than writing sql query.

It support relationship. Like pk and fk.

Create the folder as sequelize-product-api

cd sequelize-product-api

npm init -y package.json file

npm install express pg sequelize pg-hstore

express 🡪 to create rest api

pg -🡪 to connect db

sequelize🡪 provide model with function to do operation on table.

pg-hstore 🡪 model serialization and deserialization taken care by this module.

Put and patch method is use to update existing records

Put : it is use to update object all property : pname and price using id

Patch : it use to update few property or partial object. like using pid update pname or price etc.

Mkdir product\_end\_point\_mvc\_ sequelize

cd productend\_point\_using\_mvc\_style

npm install express pg sequelize pg-hstore

config -🡪 folder

db.config.js which contains database connection.

model 🡪 Folder

ProductModel.js

Which provide schema information like Product with all variable names and their data types.

Repository 🡪folder

ProductRepository.js

Which contains pure database related code.

Controller 🡪 folder

ProductController.js

Contains req and res information this function interact with repository to do operation on table.

Routes 🡪 :

ProductRoute.js

Which provide all http method information as well as sub path and base upon sub path it will call controller specific methods.

App.js or index.js or main.js

Load all modules and enable any middleware if need and provide main path for respective route file and provide port number with listener.