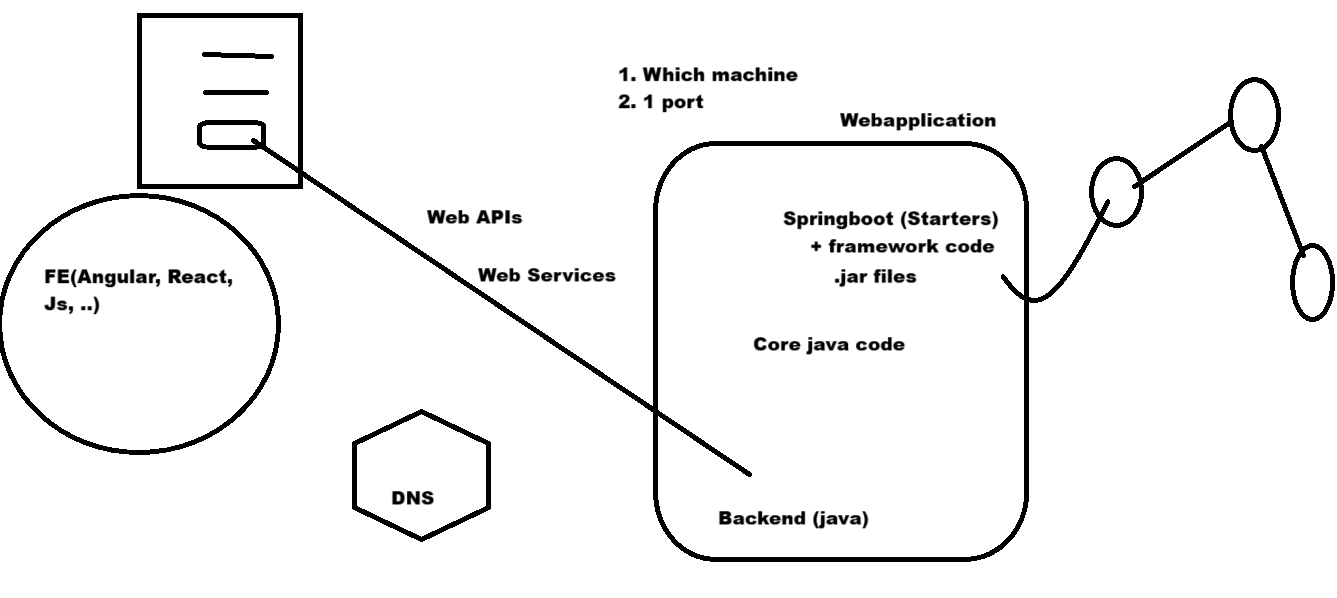
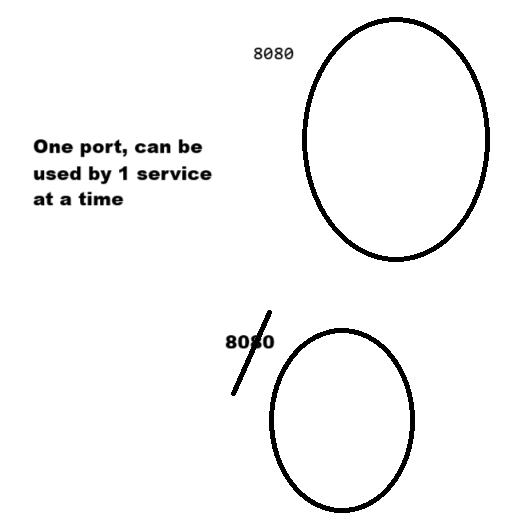
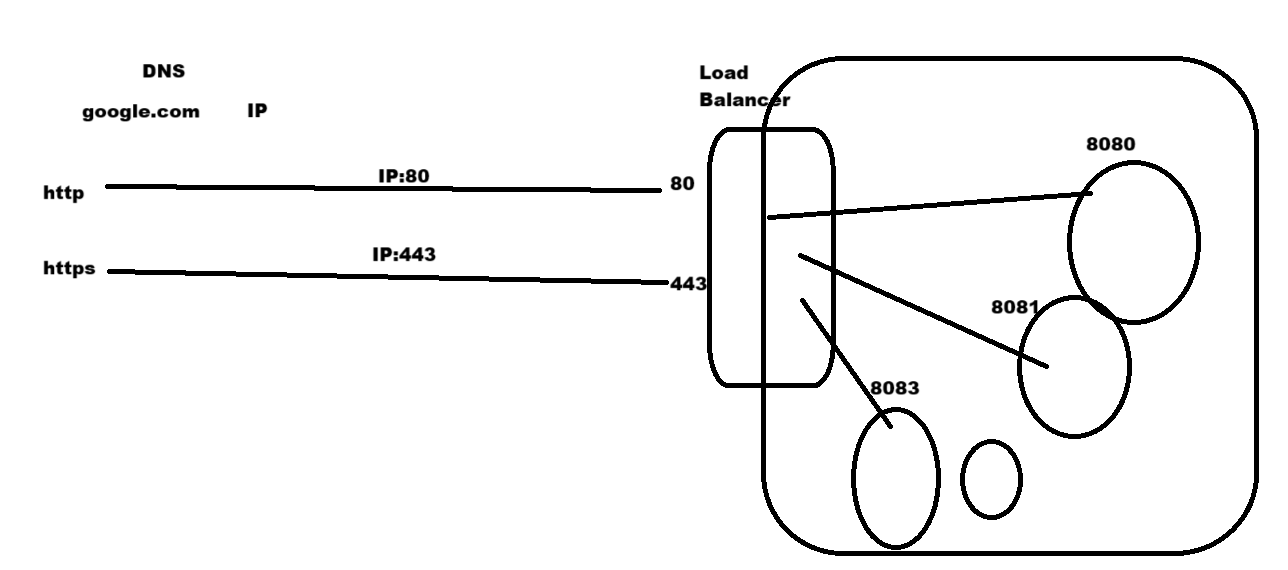
**Diagram**







**Day2: End-to-end working in IT [Technical]**

=======================

Agile

Sprints - 2weeks

finalize features

Jira

Start the sprint

Development

Web Application dev

- Java Coding (Core Java) - 10k

- Technologies (Servlets / JPS) - Web applications (6k)

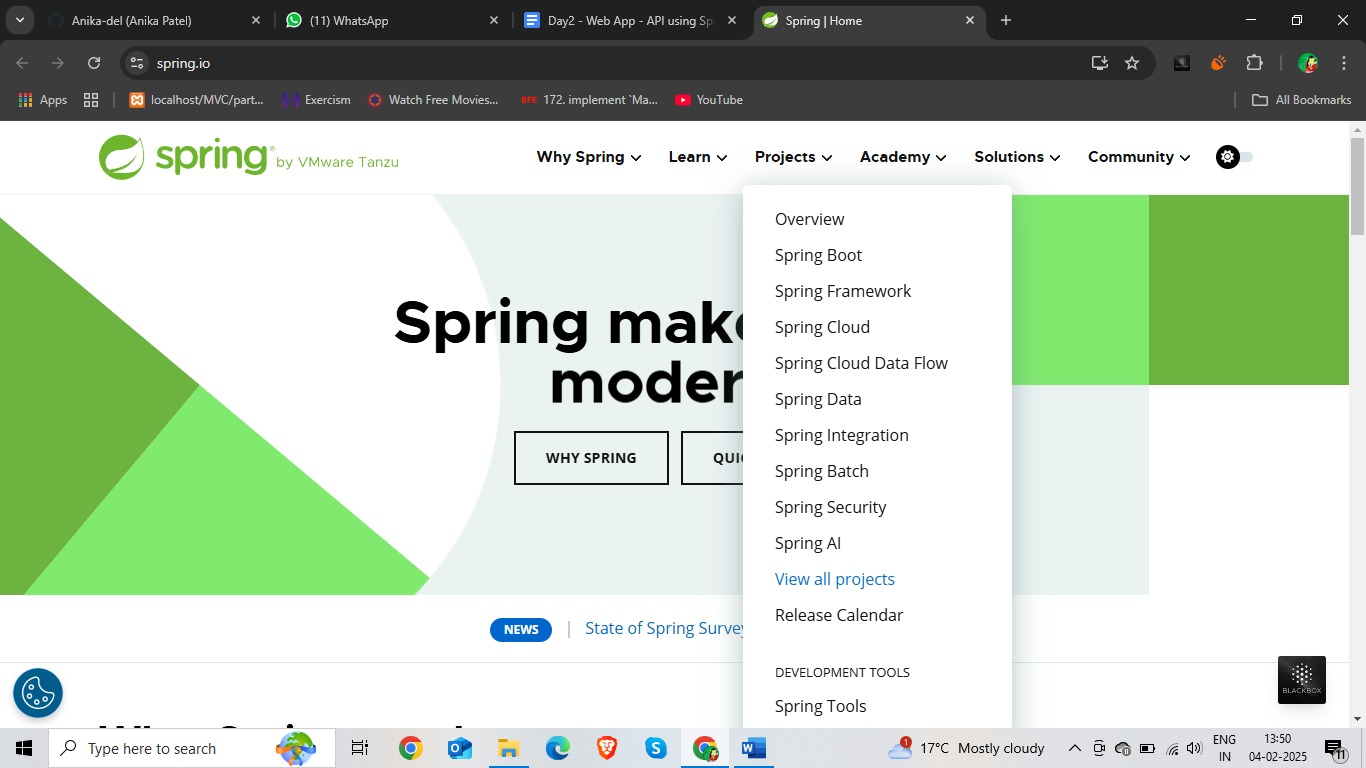
- Framework Spring

Readmade code by framework

Developer focuses on functional logic

Spring - Any kind of application

[**https://spring.io/**](https://spring.io/) **//website wisite for spring kind of application**

****

Spring Boot - one project build on top of core spring framework

It is Rapid and Fast application development.in java

Roadmap -- Explore below links:

[**https://roadmap.sh/backend**](https://roadmap.sh/backend)

[**https://roadmap.sh/java**](https://roadmap.sh/java)

[**https://roadmap.sh/spring-boot**](https://roadmap.sh/spring-boot)

//upar digram me

**Spring** & **Spring Boot** wrote code to make an application into web application, and shared that code as jar files.

Spring Boot is a powerful framework designed to simplify and speed up the development of Spring applications. Here's how some key components contribute to rapid application development:

Multiple jar hota h

Dependency management.

**Build tool: Maven, Gradle**

Spring Boot - rapid application development Powerfull application of spring boot

1).- Starters-------------

### Spring Boot Starters

Spring Boot Starters are a set of convenient dependency descriptors you can include in your application. They provide a one-stop-shop for all the dependencies you need to get started quickly with your project. By including a starter, you can avoid the hassle of manually configuring dependency versions and ensuring compatibility. Some popular starters include:

* **spring-boot-starter-web**: For building web, including RESTful, applications.
* **spring-boot-starter-data-jpa**: For accessing data using JPA.
* **spring-boot-starter-security**: For adding security to your application.

2).- Auto-configuration

### Auto-Configuration

Auto-Configuration is a powerful feature in Spring Boot that automatically configures your Spring application based on the dependencies you have added. It takes an opinionated view of the Spring platform and third-party libraries so you can get started with minimal configuration. Here's how it helps:

* **Automatic Configuration**: Based on the classpath settings, Spring Boot automatically configures the required components.
* **Conditional Configuration**: Only configures beans when certain classes are present or absent.
* **Customization**: While auto-configuration simplifies setup, it is also very flexible, allowing you to override defaults and customize settings as needed.

**Spring MVC (Frontend + Backend)** both ar running one part of system

**(UI - separated)** to another part of system **(Sever – separate)**

Communication using API (Web API)

**Many Standards to develop web API’s**

**Here are some popular web service communication protocols and architectures:**

### REST (Representational State Transfer) and gRPC (Google Remote Procedure Call)

### Comman use this time in industries

### REST (Representational State Transfer)

* **Type**: Architectural style
* **Use Cases**: Web services, APIs
* **Communication**: HTTP methods (GET, POST, PUT, DELETE)
* **Data Format**: Typically JSON or XML
* **Advantages**: Easy to use, stateless, scalable
* **Disadvantages**: Less suitable for complex queries and real-time communication

### JSON-RPC (JSON Remote Procedure Call)

* **Type**: Remote procedure call protocol
* **Use Cases**: Lightweight, simple APIs
* **Communication**: HTTP, WebSockets
* **Data Format**: JSON
* **Advantages**: Simple, lightweight, no need for strict specifications
* **Disadvantages**: Less feature-rich compared to REST or gRPC

### SOAP (Simple Object Access Protocol)

* **Type**: Protocol
* **Use Cases**: Enterprise-level web services
* **Communication**: HTTP, SMTP, TCP
* **Data Format**: XML
* **Advantages**: Strong standards, extensibility, security features
* **Disadvantages**: Verbose, complex, harder to use

### gRPC (Google Remote Procedure Call)

* **Type**: RPC framework
* **Use Cases**: Low-latency, high-performance applications
* **Communication**: HTTP/2
* **Data Format**: Protocol Buffers (binary)
* **Advantages**: High performance, bi-directional streaming, strong typing
* **Disadvantages**: Steeper learning curve, less human-readable

### GraphQL

* **Type**: Query language for APIs
* **Use Cases**: Flexible and efficient data fetching
* **Communication**: HTTP
* **Data Format**: JSON
* **Advantages**: Precise queries, reduces over-fetching, adaptable to client needs
* **Disadvantages**: Complexity in setup and security, can lead to inefficient queries if not designed well

Each of these technologies has its own strengths and ideal use cases. Whether you're building a high-performance real-time application, an enterprise-level service, or a flexible API, there's likely a perfect match among these options.

Rest API (called from outside/browser) spring boot maven (spring boot starter)

int add(int a, int b) {

return a + b;}

Faster way to build a webapplication

<https://spring.io/> //official website

<https://spring.io/quickstart> //Learn how to use in step by step

<https://spring.io/tools> //IDE-- Set up Eclipse STS

<https://start.spring.io/> //Download from here spring boot project and -Sprint Web add in this and generate

Apache Tomcat as the default embedded container.

demo.zip

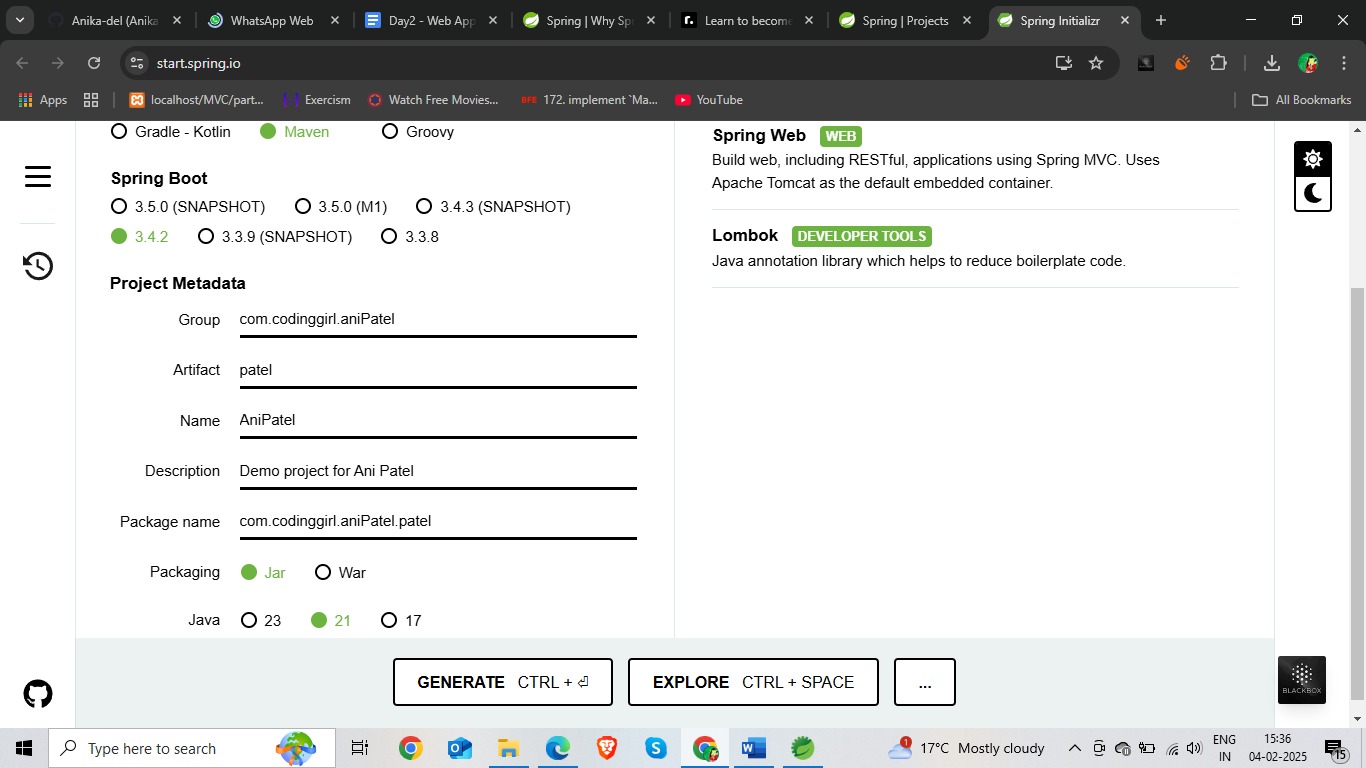
in local machine workspace

1. Set up Eclipse STS eclipse me

workspace d:\ws\ide

2. Get the demo.zip - getting started project

<https://start.spring.io/>



3. Get the demo project inside IDE

D:\ctws\feb25ct>ide copy demo unzeep and pest parallel in ide

Maven project

Extract in workspace

In IDE, Import Existing Maven Project

Eclipse>import>existimg maven project>next>root directory me pest kr do jha spring ka generate krke rkhe h >

**D:\ctws\feb25ct\temp-demo\demo**

Check pom.xml click on finish

4. explore project structure

pom.xml

5. Coding DONE.

6. How will you execute it?

Maven is along with IDE

Rt click => RunAs => Maven Build...

Open a file

clean - refresh

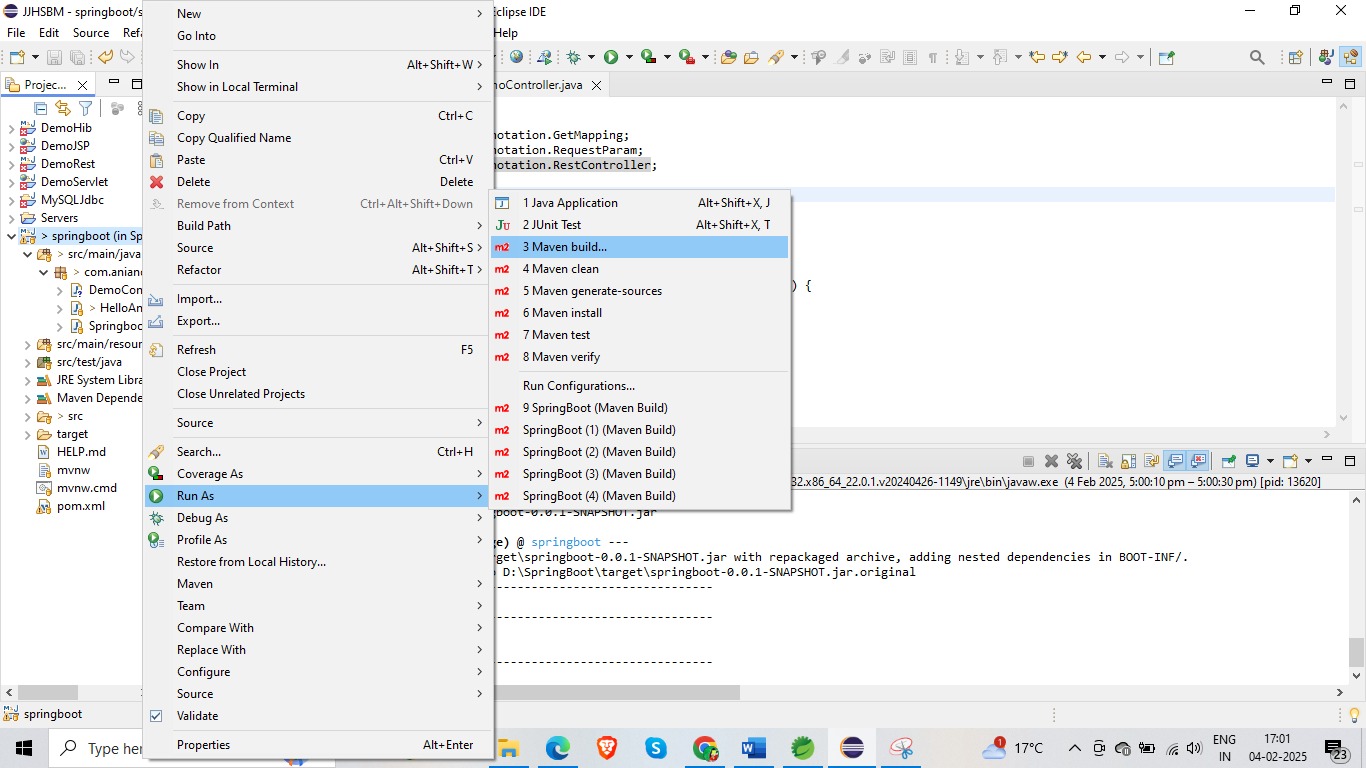
package - compile & jar

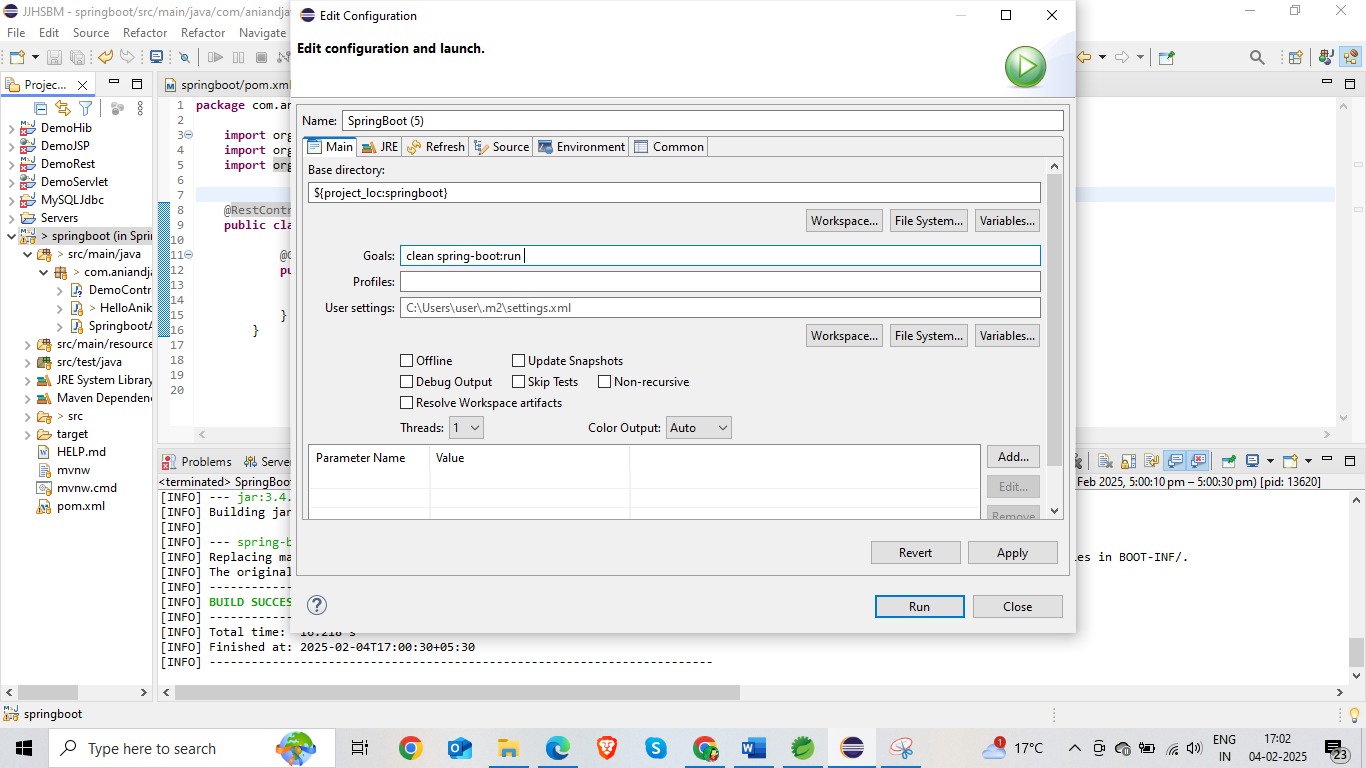
clean package dono ek sath chala sakte ho aap

run command

**clean** **spring-boot:run** - starts your spring boot application

for maven actions, it creates a "target" folder



  
--------------------------------------------------------------------------------------------------------------------------javac a.java

java a

---------------------------------------------------------------------------------------------------------------------------

for executing web application - you need server

"tomcat"

7. tomcat is default embedded server which comes along with spring web.

clean spring-boot:run

tomcat executes & spring boot app is running on

default port: 8080 **but mine 8081**

**at a time ek hi port open hoga same name se agar aapko dusra open krna h to name change krna padega**

If you try to run another service on same port, it will fail with exception.

8. How to access this service

Using url, others can access your web apis.

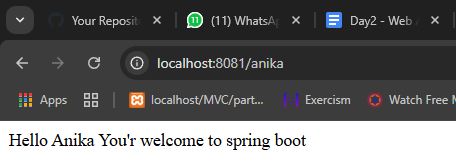
what url syntax

**http://<machine-ip>:<port>/<path>**

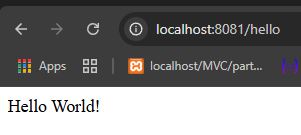
**localhost- 127.0.0.1**

[**http://localhost:8080/hello**](http://localhost:8080/hello)

**but mera port number 8081 h to ----------------** [**http://localhost:8081/anika**](http://localhost:8081/anika)

****

[**http://localhost:8081/hello**](http://localhost:8081/hello)

****

hello()

----------------------------------------------------------------------------------------------------------------

Once you start your spring boot service,

**if you make any modification,**

then

you need to

eclipse red button---- stop (click terminate)

again run as>maven build> **clean** **spring-boot:run** & start service again.

**package com.aniandjava.springboot;**

**import org.springframework.web.bind.annotation.GetMapping;**

**import org.springframework.web.bind.annotation.RequestParam;**

**import org.springframework.web.bind.annotation.RestController;**

**@RestController**

**public class DemoController {**

**@GetMapping("/hello")**

**public String hello(**

**@RequestParam(value = "name", defaultValue = "World") String name) {**

**System.out.println("Running hello() Hello " + name); //ye console pr print hoga**

**return String.format("Hello %s!", name);**

**}**

**}**

======================================================

<http://localhost:8081/hello>

you can change url according to this name put and value print

http://localhost:8080/hello?name=Feb25CT

Old Server ko band krna ho to

- Explicity kill the process running on 8080.

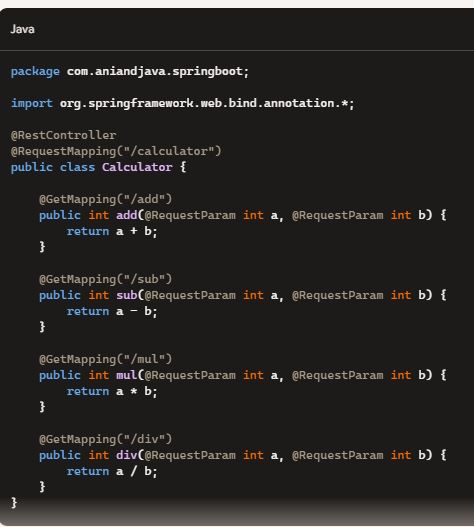
Explore using ChatGPT

netstat -ano | findstr :8080

taskkill /F /PID 12345

----------------------------------------------------------------------------------------------------------------

**CalculatorController add, sub, div, mul** for 2 integers & return the value



URLs for each endpoint, assuming your application is running on <http://localhost:8081>:

url= http://localhost:8081/calculator/add

<url>?<param1-name>=<param1-value>&<param2-name>=<param2-value>&...so on

* Add: http://localhost:8081/calculator/add?a=10&b=5
* Subtract: http://localhost:8081/calculator/sub?a=10&b=5
* Multiply: http://localhost:8081/calculator/mul?a=10&b=5
* Divide: http://localhost:8081/calculator/div?a=10&b=5

**Http Verbs**

GET, POST, PUT, DELETE, HEAD …

**Post-** Add post method in spring boot:

@PostMapping("/add-post")

    public int addPost(@RequestParam int a, @RequestParam int b) {

        return a + b;

    }

//=====not because post can’t take input from browser url

Debugging in browser:

rt-click => inspect => network tab //error ko check kr sakte h

=======================================================================

hard reload option of browser

inspect mode, long press refresh, empty cache & hard reload.

browser passes data as GET.... for postmapping method it will fail.

**How do you test all possible api calls???**

**PostMan**,....

Invoking Post API call, using Postman.

Explore all your problems via AI tools.

**- How to add new dependency. Look for the <dependency> configuration from mvnrepository.com //all dependency copy from their**

**& add to pom.xml**

clean package.

devtool maven dependency. **pom.xml**

- Have AI Pair programming - GitHub Copilot

- What is sonarqube **setup in your IDE**

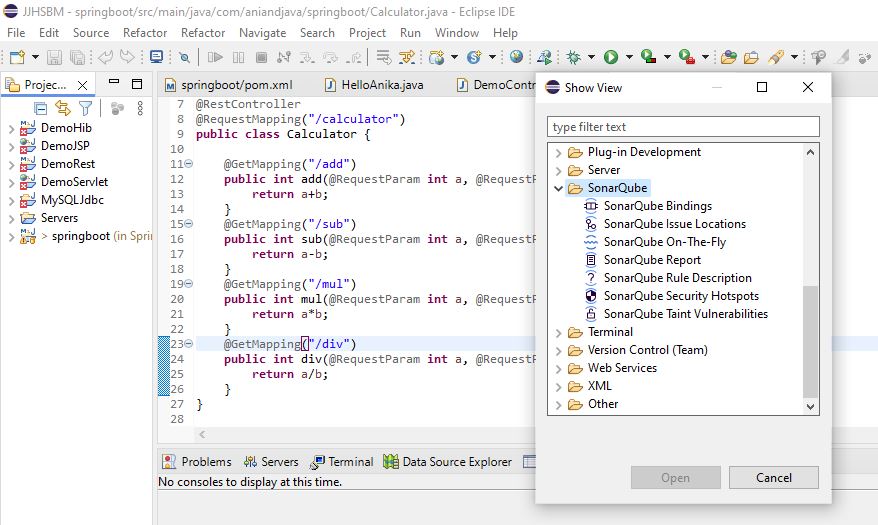
static code analyzer, best practices of coding is bundled together in sonar. it continuously checks your code, and gives you recommendations of how to improve.

How to find

Help => Eclipse market place => sonarqube

(sonarqube) install in eclipse

Windows => show view.>other> sonarqube



http and https

http://localhost:8080/calculator/add?a=5&b=10

https://google.com

https://studio.youtube.com/

DNS (Godaddy, Cloudflare, namecheap)

studio.youtube.com <IP>

https://IP:443/

http://IP:80/

------

https: 443

http: 80

------

nslookup google.com