**Setting up the development environment:**

Follow the below steps to create a sprint boot project structure

1.Navigate to the following url <https://start.spring.io/>

2.select project type as maven

3.select the programming language in which you would like to write code

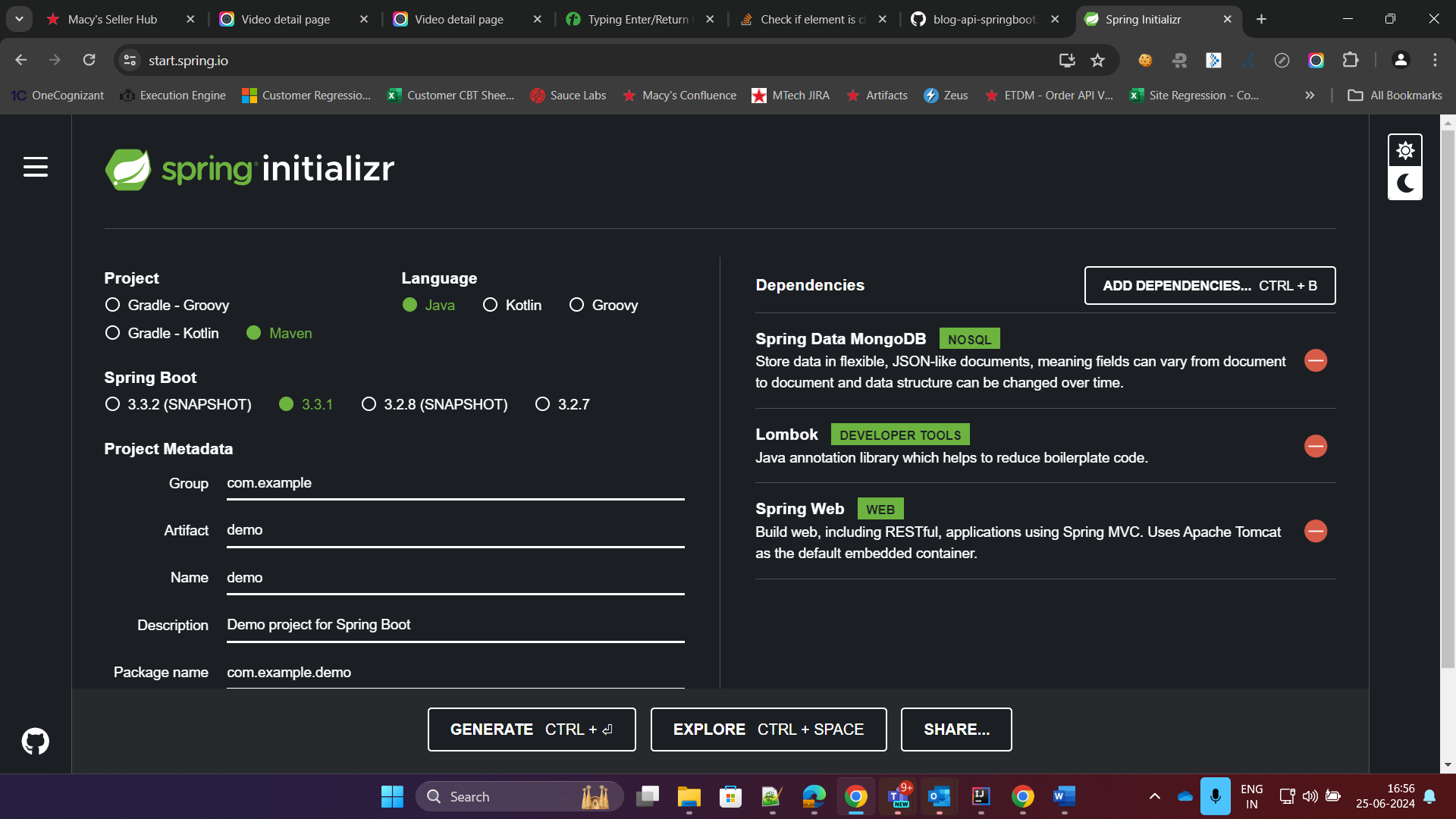
4.select spring boot version

5.under project meta data provide the following details

* Artifact name
* Name of the project
* Description
* Package name
* Java version

6.Add dependencies to your project

* Lombok
* Spring boot weblex
* Spring data mongo



And on click of generate the file will be downloaded as zip and you can extract the project file and use it in intellj or some development environment.

**Defining domain models:**

* Declare variables and their datatypes in model class
* Create a setter and getter method for the variables which you want to store the details
* Declare collection name on top of the model class

for example: @Document(collection = **"blogdetails"**)

example variables:

* blog id
* blog content
* author
* published date
* image url

Assign all variables using “this” instance keyword

**implementing repository :**

create a interface class for repository and extends using mongorepository<Model,DataType>

for Example:

public interface blogRepository extends MongoRepository<BlogDetails, String>

**service layers:**

create a service class for the actions (crud) which we are going to do on the repository

for example:

repository.save();

repository.deleteById(id);

**repository.findAll();**

* create article
* update article
* delete article
* fetch all article

**configuring security:**

**in application properties we have to define the name of the application,database,port and host details.**

**For example:**

**spring.application.name**=**Blogs  
server.port**=**8080  
spring.data.mongodb.host**=**localhost  
spring.data.mongodb.port**=**27017  
spring.data.mongodb.database**=**BlogDB**

**REST endpoints:**

**Method – endpoints – usage**

@GetMapping(path=**"api/blogId"**)

**1.Get –** [**http://localhost:8080/api/blogId**](http://localhost:8080/api/blogId) **- Retrieve blog details by Id**

@PostMapping(path=’api/blogpost)

**2.Post -** [**http://localhost:8080/api/blogPost**](http://localhost:8080/api/blogPostd) **- Add blog post**

@DeleteMapping(path=**"/api/{id}"**)

**3.Delete -** [**http://localhost:8080/api/blogId**](http://localhost:8080/api/blogId) **-Delete by Id**

@PutMapping(path=**"/api/{id}"**)

**4.Put -** [**http://localhost:8080/api/blogId**](http://localhost:8080/api/blogId) **- Update details by sending id in path varaiables and add details in body**

**5.Post -** [**http://localhost:8080/api/comments/blogId**](http://localhost:8080/api/comments/blogId) **- add comment to the blog by sending blog id and comment,date,time ,commented added by whom in the body**