System Requirements Specification Index

For

Notes-App

Version 1.0

IIHT Pvt. Ltd.

Table of Contents

1 PROJECT ABSTRACT

Note App is a FullStack Application with backend to be implemented using Spring boot with MySQL and Frontend using Angular. It allows any unregistered users (visitors) to manage the notes like create, view, modify and delete.

Visitors can perform the follow actions:

1. Allows to add a note
2. Allows to delete an existing note
3. Allows to update an existing note
4. Allows to search any note based on the id.
5. Allows to display all the notes

2 ASSUMPTIONS, DEPENDENCIES, RISKS / CONSTRAINTS

- While fetching the note by ID, if note id does not exist then the operation should throw an exception.
- While deleting the note by ID, if note id does not exist then the operation should throw an exception.
- While updating the status of note, if note id does not exist then the operation should throw an exception.
- All the database operations must be implemented on entity object only
- Do not change, add, remove any existing methods in service layer
- In Repository interfaces, custom methods can be added as per requirements.
- Must not go and touch the test resources, as they will be used for Auto-Evaluation
- All RestEndpoint methods and Exception Handlers must return data wrapped in ResponseEntity

3 REST ENDPOINTS

Rest End-points to be exposed in the controller along with method details for the same to be created

3.1 NOTECONTROLLER

URL Exposed		Purpose
/noteservice/all		Fetches all the notes
Http Method	GET	
Parameter 1	-	
Return	List<Note>	
/noteservice/add		Add a new note
Http Method	POST	
Parameter 1	Note	
Return	Note	
/noteservice/delete/{id}		Delete note with given note id
Http Method	DELETE	
Parameter 1	Integer (id)	
Return	Note	
/noteservice/get/{id}		Fetches the note with the given id
Http Method	GET	
Parameter 1	Integer (id)	
Return	Note	
/noteservice/update		Updates existing note
Http Method	PUT	
Parameter 1	Note	
Return	Note	

4 TEMPLATE CODE STRUCTURE (BACKEND-SPRING BOOT)

4.1 PACKAGE: COM.YAKSHA.ASSESSMENTS.NOTESSERVICE

Resources

NotesserviceApplication (Class)	This is the SpringBoot starter class of the application.	Already Implemented

4.2 PACKAGE: COM.YAKSHA.ASSESSMENTS.NOTESSERVICE.MODEL

Resources

Class/Interface	Description	Status

Note (class)	○ Annotate this class with proper annotation to declare it as an entity class with Id as primary key. ○ Map this class with note table. ○ Generate the Id using the IDENTITY strategy	Partially implemented.

4.3 Package: com.yaksha.assessments.notesservice.repository

Resources

Class/Interface	Description	Status
NoteRepository (interface)	1. Repository interface exposing CRUD functionality for Note Entity. 2. You can go ahead and add any custom methods as per requirements	Partially implemented

4.4 Package: com.yaksha.assessments.notesservice.service

Resources

Class/Interface	Description	Status
NoteService (interface)	Interface to expose method signatures for note related functionality. Do not modify, add or delete any method	Already implemented.
NoteServiceImpl (class)	● Implements NoteService. Contains template method implementation.	To be implemented.

	● Need to provide implementation for note related functionalities ● Do not modify, add or delete any method signature	

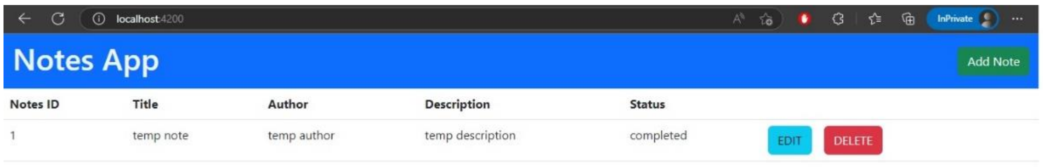
4.5 PACKAGE: COM.YAKSHA.ASSESSMENTS.NOTESSERVICE.CONTROLLER

Resources

Class/Interface	Description	Status
NoteController (Class)	● Controller class to expose all rest-endpoints for note related activities. ● May also contain local exception handler methods	To be implemented

5 PROPOSED WIREFRAME (FRONTEND-ANGULAR)

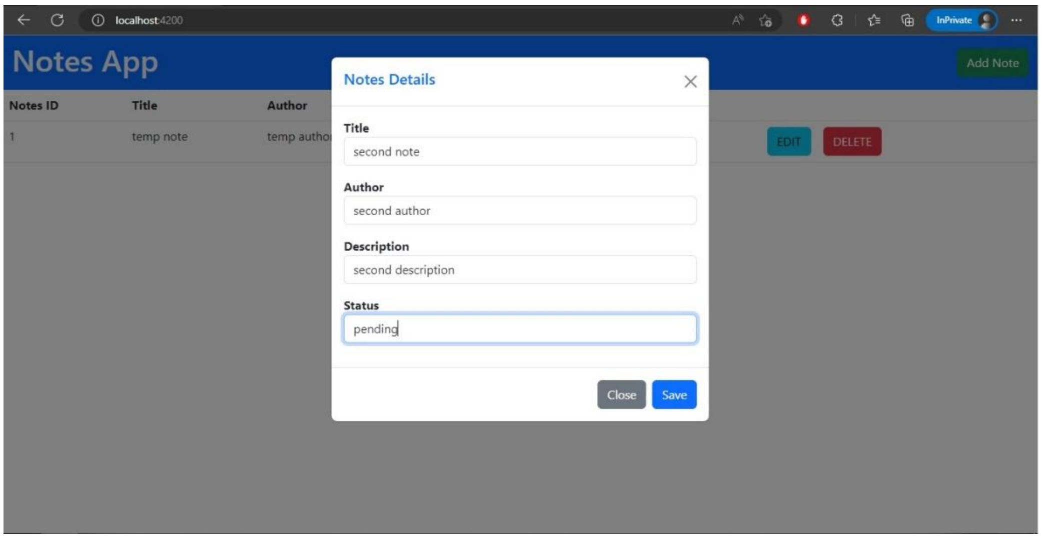
5.1 LISTING ALL THE NOTES



Notes App

Notes ID	Title	Author	Description	Status	
1	temp note	temp author	temp description	completed	EDIT DELETE

5.2 Adding a new note



localhost:4200

Notes App

Add Note

Notes ID	Title	Author
1	temp note	temp autho

Notes Details

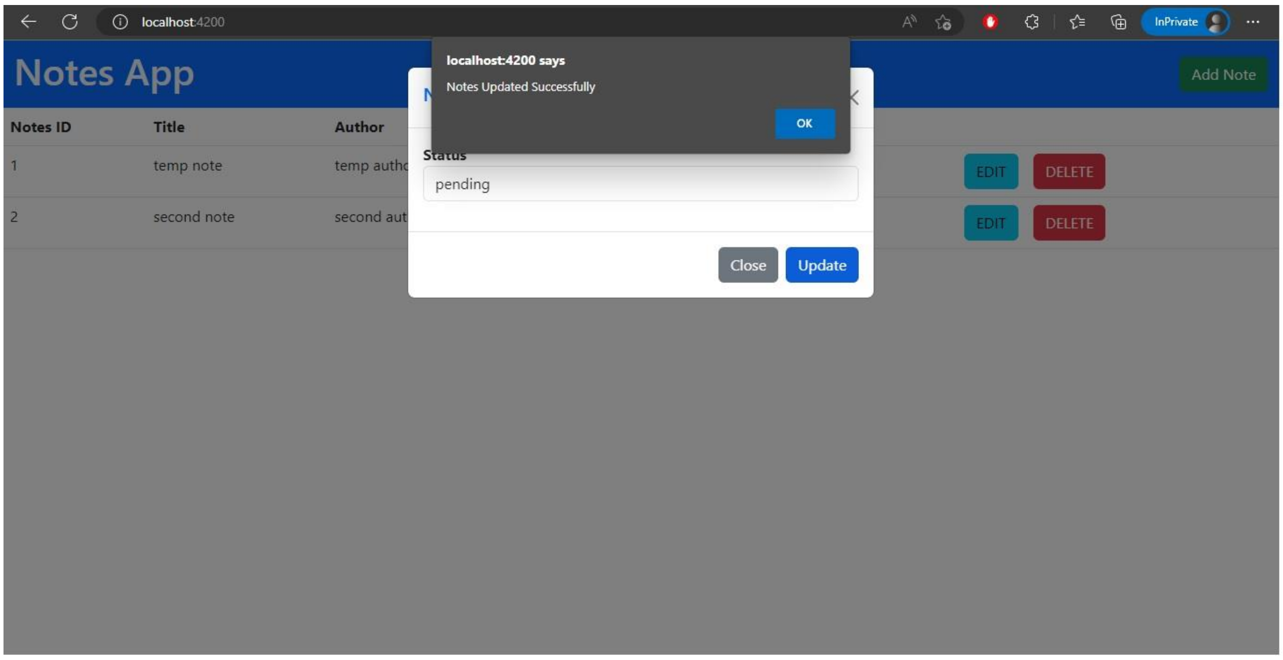
Title: second note

Author: second author

Description: second description

Status: pending

5.3 Updating the status of note



localhost:4200

Notes App

Add Note

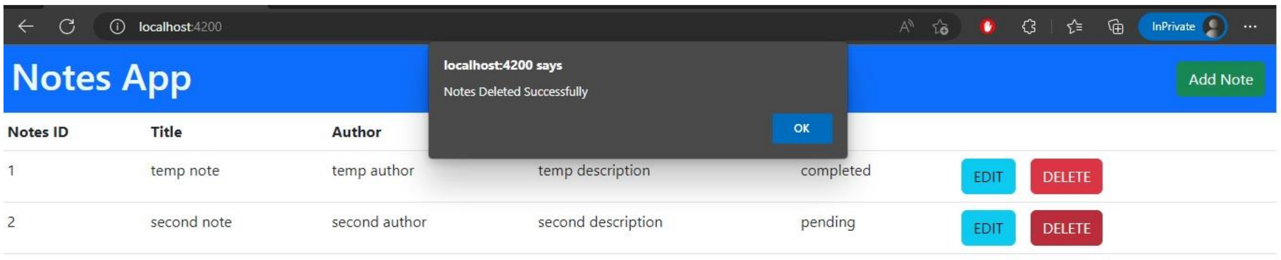
Notes ID	Title	Author
1	temp note	temp autho
2	second note	second aut

localhost:4200 says

Notes Updated Successfully

Status: pending

5.4 DELETING THE NOTE



localhost:4200 says

Notes Deleted Successfully

Notes ID	Title	Author			
1	temp note	temp author	temp description	completed	EDIT DELETE
2	second note	second author	second description	pending	EDIT DELETE

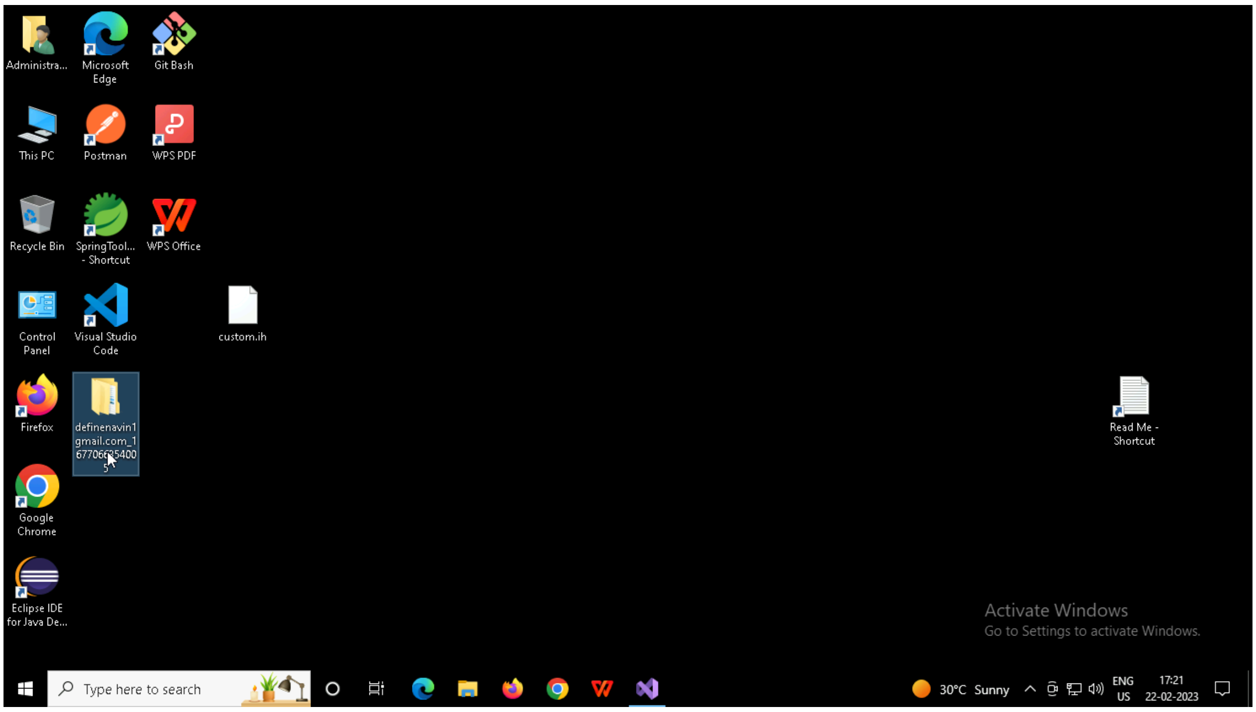
6 EXECUTION STEPS TO FOLLOW

1. Import the Java (Spring boot) project in Eclipse IDE.
2. You can build, run and test the project using IDE tools
3. README file on desktop contains credentials to access MySQL
4. Open the Angular project in VS Code.
5. You are required to install Angular package libraries before you start working on Angular Project. Command: npm install
6. To run the Angular project use command: npm start

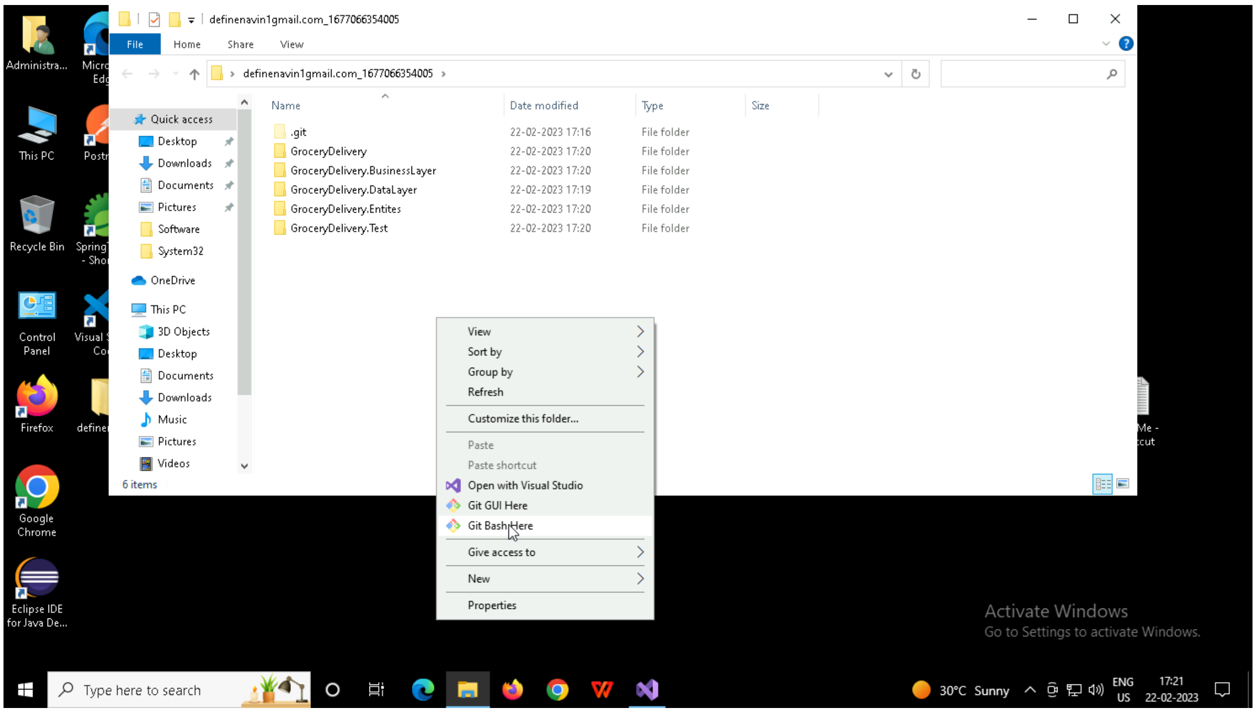
7 MANDATORY STEPS TO FOLLOW

1. **You are mandatory required to run test cases for both the applications before final submission. Without which project evaluation will not happen**
2. **You can run the Junit test cases using Eclipse menu options**
3. **To run Angular test cases, use command: npm test**
4. **Before final submission, you are also required to push your code to GIT. Following are the steps to follow:**

Open the project folder available on desktop



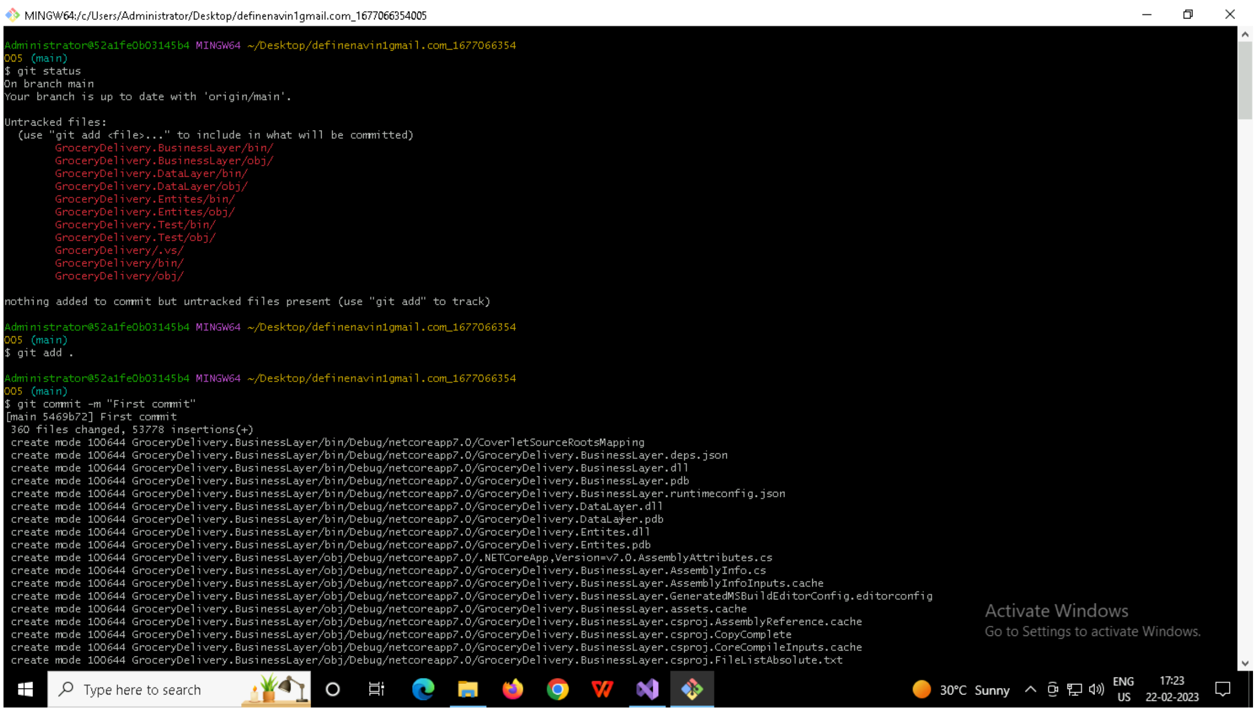
Right click in folder and open Git Bash



Name	Date modified	Type	Size
.git	22-02-2023 17:16	File folder	
GroceryDelivery	22-02-2023 17:20	File folder	
GroceryDelivery.BusinessLayer	22-02-2023 17:20	File folder	
GroceryDelivery.DataLayer	22-02-2023 17:19	File folder	
GroceryDelivery.Entites	22-02-2023 17:20	File folder	
GroceryDelivery.Test	22-02-2023 17:20	File folder	

In Git bash terminal, run following commands

b. git status
c. git add .
d. git commit -m “First commit”
   (You can provide any message every time you commit)
e. git push


MINGW64:/c/Users/Administrator/Desktop/definenavin1gmail.com_1677066354005

```
Administrator@52a1fe0b03145b4 MINGW64 ~/Desktop/definenavin1gmail.com_1677066354
005 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        GroceryDelivery.BusinessLayer/bin/
        GroceryDelivery.BusinessLayer/obj/
        GroceryDelivery.DataLayer/bin/
        GroceryDelivery.DataLayer/obj/
        GroceryDelivery.Entites/bin/
        GroceryDelivery.Entites/obj/
        GroceryDelivery.Test/bin/
        GroceryDelivery.Test/obj/
        GroceryDelivery/.vs/
        GroceryDelivery/bin/
        GroceryDelivery/obj/

nothing added to commit but untracked files present (use "git add" to track)

Administrator@52a1fe0b03145b4 MINGW64 ~/Desktop/definenavin1gmail.com_1677066354
005 (main)
$ git add .

Administrator@52a1fe0b03145b4 MINGW64 ~/Desktop/definenavin1gmail.com_1677066354
005 (main)
$ git commit -m "First commit"
[main 5469b72] First commit
 360 files changed, 53778 insertions(+)
 create mode 100644 GroceryDelivery.BusinessLayer/bin/Debug/netcoreapp7.0/CoverletSourceRootsMapping
 create mode 100644 GroceryDelivery.BusinessLayer/bin/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.deps.json
 create mode 100644 GroceryDelivery.BusinessLayer/bin/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.dll
 create mode 100644 GroceryDelivery.BusinessLayer/bin/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.pdb
 create mode 100644 GroceryDelivery.BusinessLayer/bin/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.runtimeconfig.json
 create mode 100644 GroceryDelivery.BusinessLayer/bin/Debug/netcoreapp7.0/GroceryDelivery.DataLayer.dll
 create mode 100644 GroceryDelivery.BusinessLayer/bin/Debug/netcoreapp7.0/GroceryDelivery.DataLayer.pdb
 create mode 100644 GroceryDelivery.BusinessLayer/bin/Debug/netcoreapp7.0/GroceryDelivery.Entites.dll
 create mode 100644 GroceryDelivery.BusinessLayer/bin/Debug/netcoreapp7.0/GroceryDelivery.Entites.pdb
 create mode 100644 GroceryDelivery.BusinessLayer/obj/Debug/netcoreapp7.0/.NETCoreApp,Version=v7.0.AssemblyAttributes.cs
 create mode 100644 GroceryDelivery.BusinessLayer/obj/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.AssemblyInfo.cs
 create mode 100644 GroceryDelivery.BusinessLayer/obj/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.AssemblyInfoInputs.cache
 create mode 100644 GroceryDelivery.BusinessLayer/obj/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.GeneratedMSBuildEditorConfig.editorconfig
 create mode 100644 GroceryDelivery.BusinessLayer/obj/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.assets.cache
 create mode 100644 GroceryDelivery.BusinessLayer/obj/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.csproj.AssemblyReference.cache
 create mode 100644 GroceryDelivery.BusinessLayer/obj/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.csproj.CopyComplete
 create mode 100644 GroceryDelivery.BusinessLayer/obj/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.csproj.CoreCompileInputs.cache
 create mode 100644 GroceryDelivery.BusinessLayer/obj/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.csproj.FileListAbsolute.txt
```