System Requirements Specification Index

For

Tutor Finder Application

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Contents

1 BUSINESS-REQUIREMENT:

1.1 PROBLEM STATEMENT:

The Tutor Finder Application is designed to provide a platform to find all tutors for all courses and facilitate the connection between students and educational tutors.

You are tasked with creating a system that provides access to a home page and enables the addition of new tutors with details such as name, subject, description, duration, and price. The application should allow users to include functionalities to edit and delete tutor profiles. Additionally, there should be a search feature enabling users to find tutors by their names, thereby making the process of finding the right educational support straightforward and efficient.

1.1.1 Tutor Finder Application:

The Tutor Finder Application allows you to:

1. Access the home page.
2. Should be able to add a new tutor.
3. It can have basic fields like name, subject, description, duration and price.
4. Should be able to get the list of tutors along with options to sort in ascending and descending order in each field.
5. Should be able to edit and delete any tutor.
6. Should be able to search for a tutor by tutor name.

2. Template Code Structure:

2.1 Tutor Controller

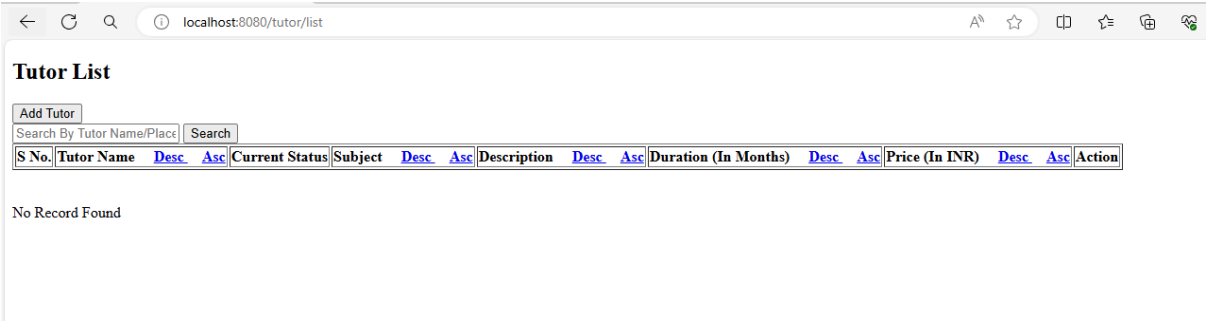
Method Exposed	Purpose
listTutors()	Should return page “list-tutors” with required data.
showFormForAdd()	Should return page “add-tutor-form” for adding a tutor.
saveTutor()	Should save a tutor and return “tutor/list” with required data.
showFormForUpdate()	Should show tutor details in page “update-tutor-form” to edit a tutor.
deleteTutor()	Should delete a tutor and return “tutor/list” with required data.
searchTutors()	Should search for a tutor and return “list-tutor” with required data.
updateTutorAvailability()	Should show current status of tutors availability

2.2 Resources Available:

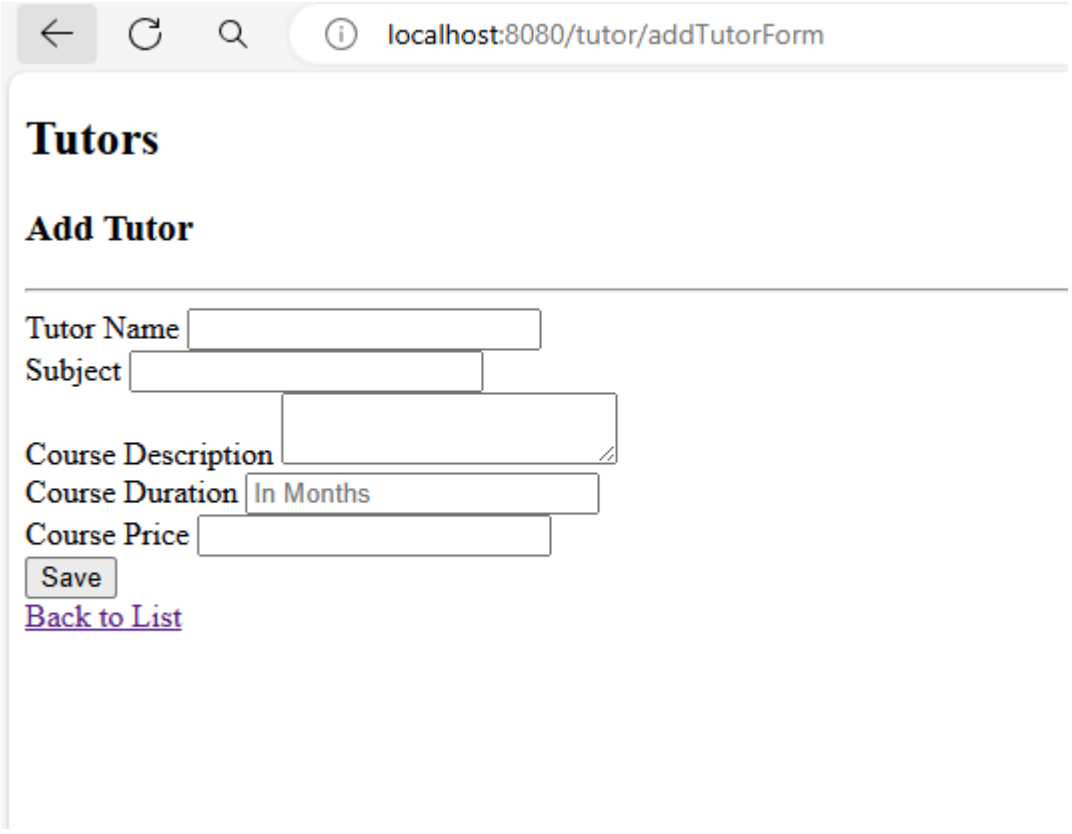
Description	View Pages Name	Remarks
Common UI		
Home Page	list-tutors	Contains a homepage which shows a list of all tutors along with options to add, edit , delete and search a tutor.
All tutor	list-tutors	
Add a tutor	add-tutor-form	
Update a tutor	update-tutor-form	
Search a tutor	list-tutors	

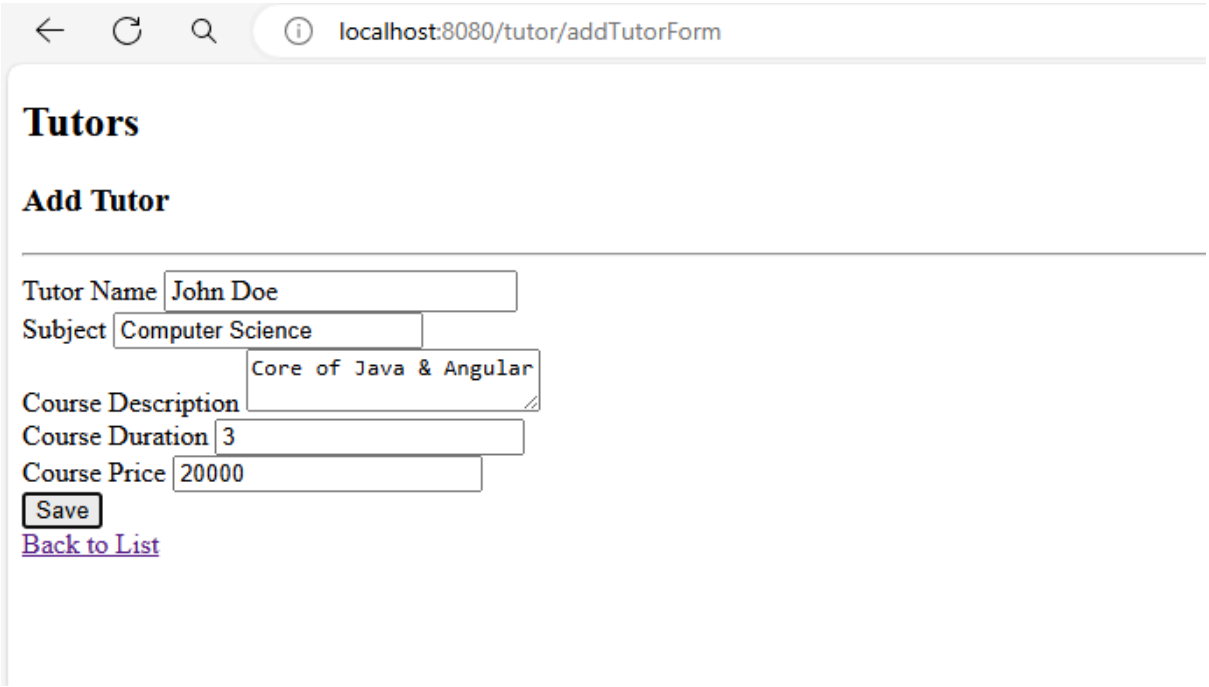
3 SUGGESTED WIREFRAMES:

1. Homepage – Visitor Landing Page

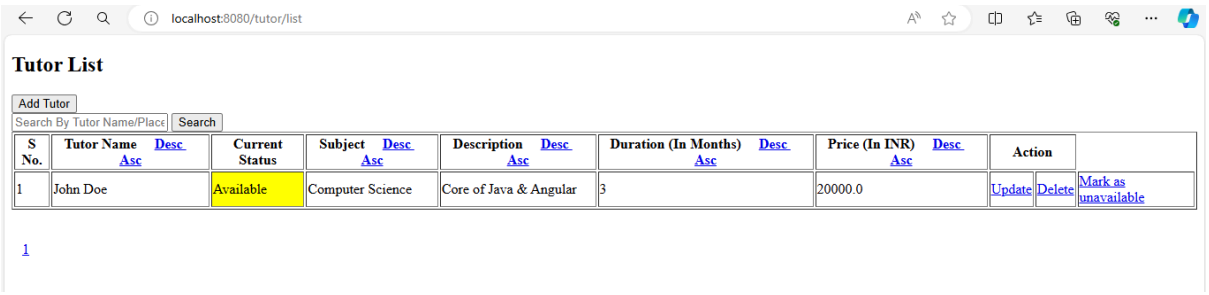


2. Create/Add a Tutor

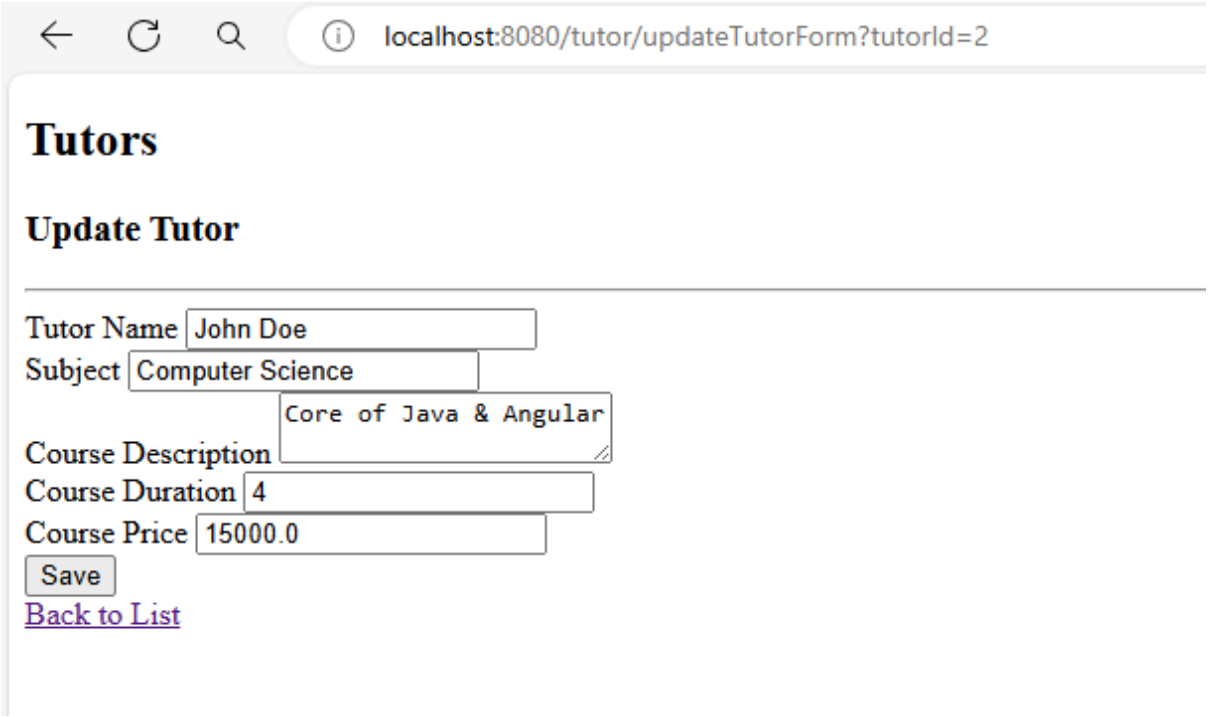


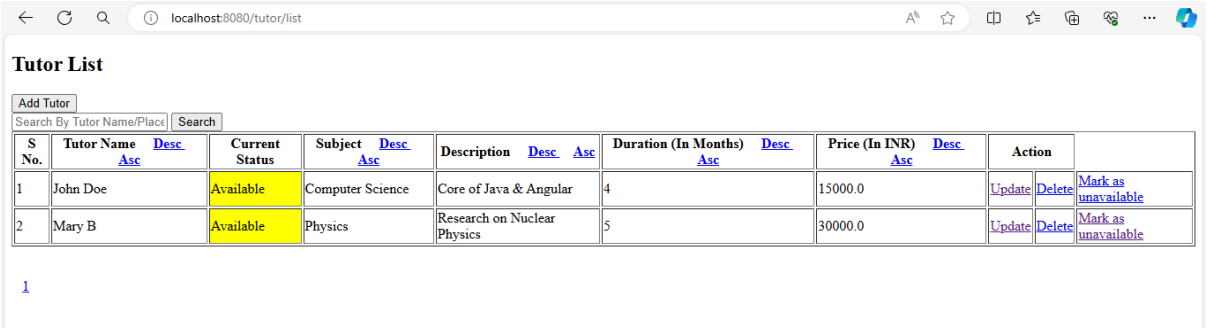

localhost:8080/tutor/addTutorForm
Tutors
Add Tutor
Tutor Name John Doe
Subject Computer Science
Course Description Core of Java & Angular
Course Duration 3
Course Price 20000
Save
Back to List

3. Tutor List

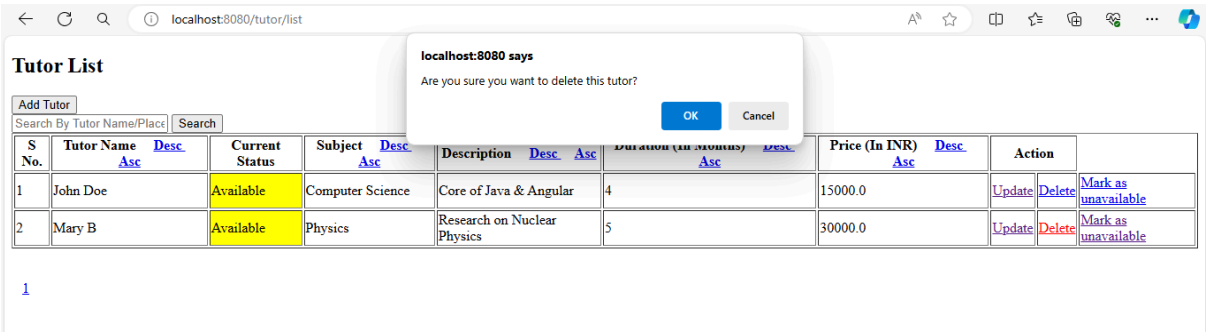

localhost:8080/tutor/list
Tutor List
Add Tutor
Search By Tutor Name/Place
Search
S No.
Tutor Name Desc Asc
Current Status
Subject Desc Asc
Description Desc Asc
Duration (In Months) Desc Asc
Price (In INR) Desc Asc
Action
1
John Doe
Available
Computer Science
Core of Java & Angular
3
20000.0
Update
Delete
Mark as unavailable
1

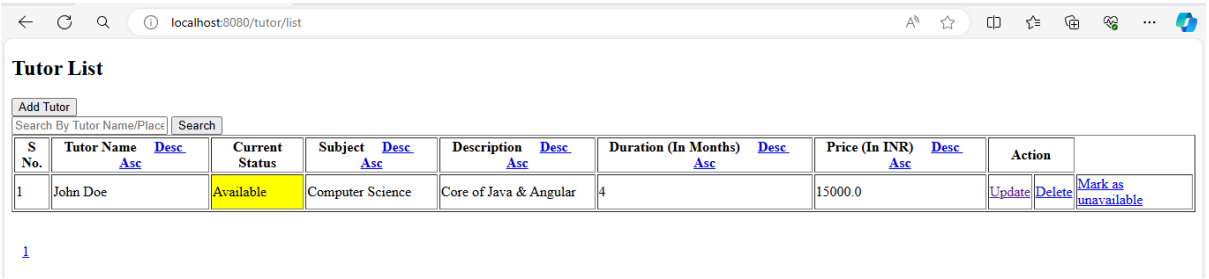
4. Update a Tutor



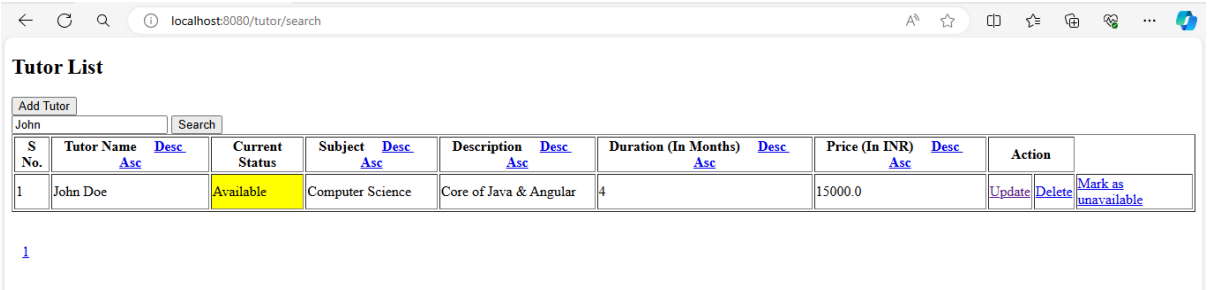


5. Delete a Tutor

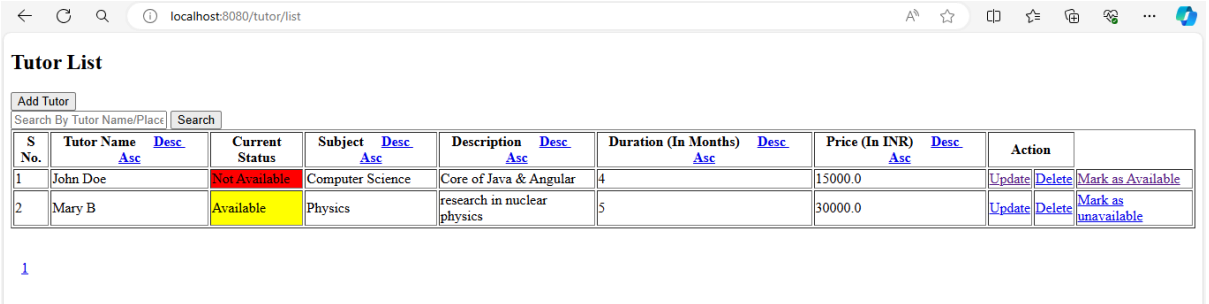




6. Search a Tutor



7. Tutor Availability



4 Business Validations

1. Id must be of type id.
2. Name value should not be blank, min 2 and max 40 characters.
3. Subject value should not be blank.
4. Description value should not be blank, min 2 and max 200 characters.
5. Duration value is not null.
6. Price value is not null.

5 Considerations

The Code template already contains skeleton methods for service and controller layer. Please write your logic in it.

6 Execution Steps to Follow

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**
2. **To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal**
3. **To build your project use command:**
   mvn clean package -Dmaven.test.skip
4. **To launch your application:**
   java -jar <your application war file name>
5. **This editor Auto Saves the code**
6. **If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.**
7. **These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.**
8. **To test any Restful application, the last option on the left panel of IDE, you can find ThunderClient, which is the lightweight equivalent of POSTMAN.**
9. **This is a web-based application, to run the application on a browser, use the**

internal browser in the workspace. Click on the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.

Note: The application will not run in the local browser

10. Default credentials for MySQL:
    a. Username: root
    b. Password: pass@word1
11. To login to mysql instance: Open new terminal and use following command:
    a. sudo systemctl enable mysql
    b. sudo systemctl start mysql

    NOTE: After typing the second sql command (sudo systemctl start mysql), you may encounter a warning message like :

    System has not been booted with systemd as init system (PID 1). Can't operate. Failed to connect to bus: Host is down

    >> Please note that this warning is expected and can be disregarded. Proceed to the next step.

    c. mysql -u root -p

    The last command will ask for password which is 'pass@word1'
12. Mandatory: Before final submission run the following command:

    mvn test
13. You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.