
System Requirements Specification Index

For

Personal Blog Application

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Contents

1	Business-Requirement:	3
1.1	Problem Statement:	3
1.1.1	Personal Blog Application	3
2.	Template Code Structure	
1.1	Blog Controller	
3.	Resources AVAILABLE:	4
4	Suggested WIREFRAMES:	5
5	Business Validations	9
6	Considerations	9
7	Execution Steps to Follow	9

1 BUSINESS-REQUIREMENT:

1.1 PROBLEM STATEMENT:

The purpose of this application is to allow a person to manage all his/her blogs.

1.1.1 Personal Blog Application:

The **Personal Blog Application** allows you to:

1. Access home page
2. Should be able to add a new blog
3. It can have basic fields like Name, Title & Description
4. Should be able to get the list of blogs.
5. Should be able to edit and delete any blog.

2. TEMPLATE CODE STRUCTURE:

2.1 BLOG CONTROLLER

Method Exposed	Purpose
getForm()	Should return page "form" for adding a blog.
submitForm()	Should submit a new blog and return page "blogs"
getBlogs()	Should return page "blog_list" with required data.
deleteBlog()	Should delete the blog and return "blog_list" page.

3. RESOURCES AVAILABLE:

Description	View Pages Name	Remarks
Common UI		
Home Page	form	Contains homepage which shows list of all blogs along with options to add, edit and delete a blog.
All Blogs	blog_list	
Update a blog	form	

3 SUGGESTED WIREFRAMES:

1. Homepage – Visitor Landing Page



A wireframe of a web form for adding a personal blog. The form is displayed in a browser window with the address bar showing 'localhost:8081'. At the top, there are two links: 'Add Blog' and 'Personal Blogs'. Below these links is the title 'Personal Blog Form'. The form contains three input fields: 'Name', 'Title', and 'Description'. Each field is preceded by a label. At the bottom of the form is a 'Submit' button.

← ↻ ⓘ localhost:8081

[Add Blog](#) [Personal Blogs](#)

Personal Blog Form

Name :

Title :

Description :

2. Create a Blog

A screenshot of a web browser window. The address bar shows 'localhost:8081'. The page content includes a link 'Add Blog Personal Blogs', a title 'Personal Blog Form', and three input fields labeled 'Name', 'Title', and 'Description'. Each field contains a sample value ('new blog', 'new title', 'new description') and is followed by a 'Submit' button.

[Add Blog Personal Blogs](#)

Personal Blog Form

Name :

Title :

Description :

3. Edit a Blog

The screenshot shows a web browser window with the address bar displaying 'localhost:8081/?id=1'. The page content includes a navigation link 'Add Blog Personal Blogs' in blue text. Below this is a heading 'Personal Blog Form' in bold black text. The form consists of three labeled input fields: 'Name : New Blog', 'Title : First', and 'Description : Description of first blog'. At the bottom of the form is a 'Submit' button.

← ↻ ⓘ localhost:8081/?id=1

[Add Blog Personal Blogs](#)

Personal Blog Form

Name :

Title :

Description :

4. All Blogs



BUSINESS VALIDATIONS

1. Id must be of type id.
2. Name value not blank, min 3 and max 20 characters.
3. Title value not blank, min 3 and max 20 characters.
4. Description value not blank and min 3 and max 200 characters.

4 CONSIDERATIONS

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

5 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 springboot-personal-blog-0.0.1-SNAPSHOT.war
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.