**The Online Survey System**

**Microservice**

**SOFTWARE DEVELOPMENT REQUIREMENT AND PLANNING DOCUMENT**

**Requirements to develop the Micro service**

* IntelliJ IDE

IntelliJ IDEA is an integrated development environment (IDE) written in Java for developing computer software. It is developed by JetBrains (formerly known as IntelliJ), and is available as an Apache 2 Licensed community edition, and in a proprietary commercial edition.

* Maven dependency

Maven is a project management and comprehension tool that provides developers a complete build lifecycle framework. Development team can automate the project's build infrastructure in almost no time as Maven uses a standard directory layout and a default build lifecycle.

* H2 database

H2 is a **relational database management system** written in Java. It can be embedded in Java applications or run-in client-server mode. The software is available as open-source software Mozilla Public License 2.0 or the original Eclipse Public License.

* Junit dependency

JUnit is a unit testing framework for the Java programming language. JUnit has been important in the development of test-driven development, and is one of a family of unit testing frameworks which is collectively known as x Unit that originated with S Unit.

* JDK 1.8

The **JDK** is a development environment for building applications, applets, and components using the Java programming language. The **JDK** includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform

* Spring Boot Framework

**Spring** **Boot** is an open-source Java-based **framework** used to create a micro-Service. It is developed by Pivotal Team and is used to build stand-alone and production ready spring applications.

* Thyme leaf

**Thyme leaf** is a modern server-side Java template engine for both web and standalone environments.**Thyme leaf’s** main goal is to bring elegant natural templates to your development workflow — HTML that can be correctly displayed in browsers and also work as static prototypes, allowing for stronger collaboration in development teams.

* Html

The**Hypertext Markup** Language, or HTML is the standard**markup** language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

* CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

* Java Script

JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

* Google charts API

Google Charts is an interactive Web service that creates graphical charts from user-supplied information. The user supplies data and a formatting specification expressed in JavaScript embedded in a Web page; in response the service sends an image of the chart.

* Bootstrap

**Bootstrap** is a free front-end framework for faster and easier web development **Bootstrap** includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins

* jQuery

jQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax. It is free, open-source software using the permissive MIT License.

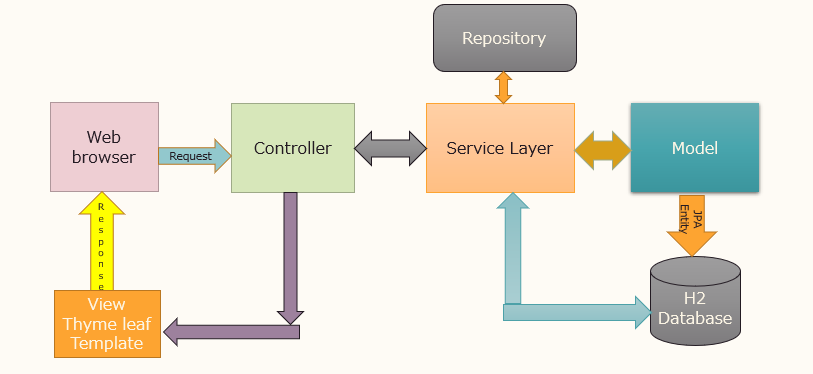
* Canva

**Canva** is a [graphic design](https://en.wikipedia.org/wiki/Graphic_design) platform, used to create [social media](https://en.wikipedia.org/wiki/Social_media) graphics, [presentations](https://en.wikipedia.org/wiki/Presentations), [posters](https://en.wikipedia.org/wiki/Posters), [documents](https://en.wikipedia.org/wiki/Documents) and other visual content.[[3]](https://en.wikipedia.org/wiki/Canva#cite_note-techcrunch2-3)[[4]](https://en.wikipedia.org/wiki/Canva#cite_note-4)[[5]](https://en.wikipedia.org/wiki/Canva#cite_note-5) The app includes templates for users to use. The platform is free to use and offers paid subscriptions like Canva Pro and Canva for Enterprise for additional functionality.

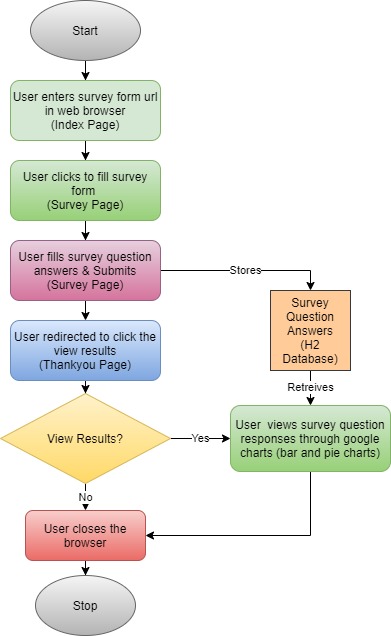
**STATEMENT OF PURPOSE**

* Online Survey System is a micro service which conducts Covid 19 return to work survey and stores all the participants responses with all questions in the H2 database. It also displays the participants’ answers and the overall results in different charts. So, it is easy to get an idea about people’s interests.
* This microservice displays the interactive and pleasing web page to attract the viewers to take part in the survey. It stores the participants answers in the database and the results for each question and the overall results will be available immediately for the participants to know others people’s thoughts and how other people have answered.
* Covid 19 Return to work survey helps people to express their interests to return to work and assessed based on different questions on different areas. It also helps the organizations to take important decisions after seeing the chart results.

**Functionality Diagram**



**The Online Survey System PROCESS FLOW**

****

**LIST OF FEATURES**

1. The Online Survey System Microservice starts with the stunning webpage with the URL <http://localhost:8082>/ with lots of styles added in it to attract the viewers to participate in this survey.
2. Hyperlink is provided to navigate to the second page <http://localhost:8082>/survey.
3. Second page contains all the survey questions with the radio buttons for the participants to log their entries easily.
4. At the end of this page ‘submit’ button is provided to submit the form once it is finished.
5. This page also has lots of attractive styles including the addition of panels, colors, attractive submit button and more additional styles.
6. Once the form is submitted all the questions along with the responses goes straight to the H2 Database and get stored there.
7. In the database auto generated row Id is provided to know the number of entries easily.
8. Also, current date and time is provided to know when the entries are logged into the H2 database.
9. Current date and time are also formatted using Date Time Formatter class.
10. After clicking the submit button it navigates to the third page <http://localhost:8082/thankyou>. This page says thankyou to all the participants who has taken part in the survey with an attractive image added to this page.
11. This thankyou page also a hyperlink “Click here to see the overall results” that takes us to the fourth and the last page of this microservice <http://localhost:8082/result> .
12. The result page makes it easy to understand the survey results immediately using Google chart API.
13. Column chart and pie chart are provided to better understand the trend of this survey and the interests of the people that makes individuals and organizations to take vital decisions efficiently.

**IMPLEMENTATION PLAN**

1. First, I started with creating a new repository on GitHub and named it The Online Survey System.
2. I created a Spring boot project using spring initializer and added all the required dependencies (Spring Boot, Maven and Thyme leaf) and started using this project in IntelliJ IDE. Also, junit dependency has been added by changing the properties suitable for my project in pom.xml.
3. I cloned the GitHub Repo in to my machine and made a local repo and added the new Spring Boot project in to it.
4. I gave my first commit and pushed it from IntelliJ.
5. I added the database details and the port details in application. Properties to set up the database and checked whether the database gets connected by using the DB URL.
6. Then I Started thinking for my source code. At this stage I was researching and planning for the implementation.
7. I was planning to show a list of questions for the participants. So, I created a class Survey Details with the survey variables, constructor and getters and setters. This serves the model for this microservice.
8. Repository layer has been created and Survey Details Repository class has been added with Jpa repository.
9. Service layer has been added and inside this package Survey Service Interface and Survey impl Class has been added to implement Survey service interface.
10. Web Controller layer has been created and the Survey controller class has been added and I was planning to have all the logic in it.
11. I started coding for the index page using HTML and Thyme leaf. Get mapping (“/”) was done with welcome page () method in the controller to access the webpage.
12. Then I made it run by main () method.
13. Then slowly I created the following HTML pages

* Survey.html
* Thankyou.html
* Result.html

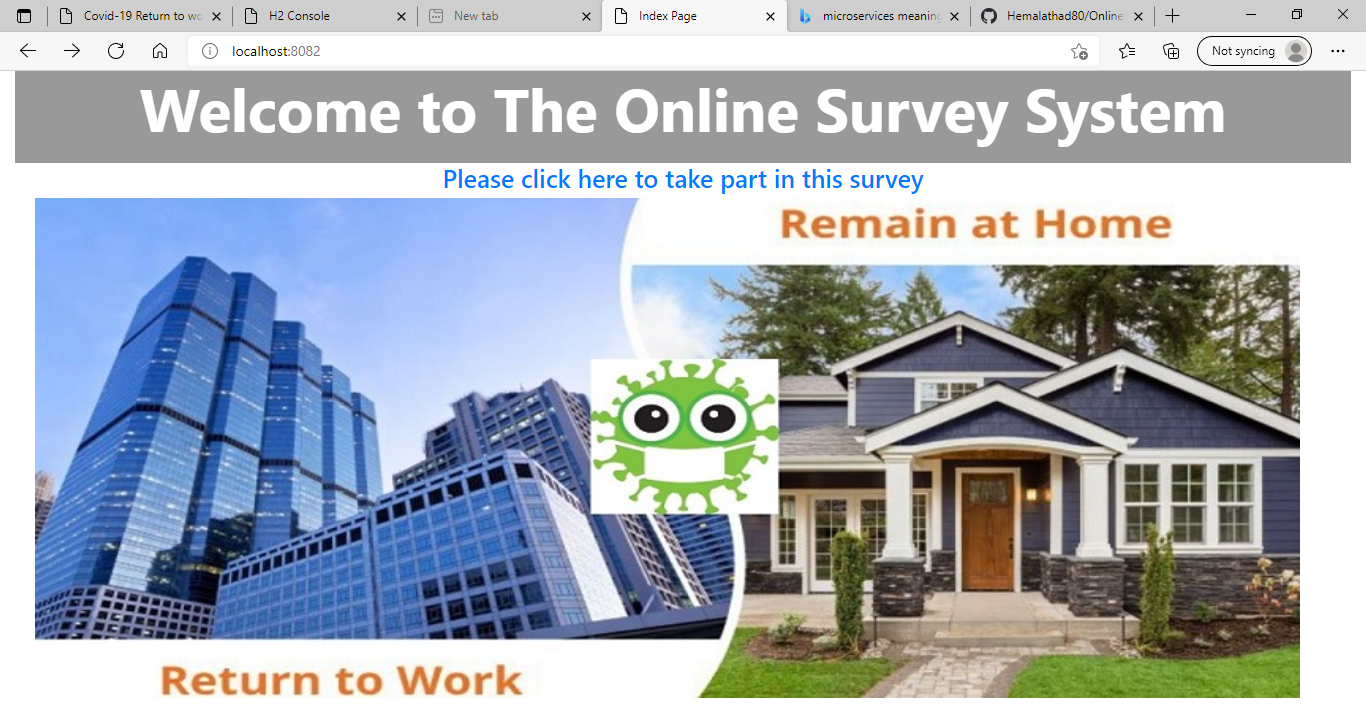
1. In the controller class I created a list of questions using List in Java that maps the survey page with Get mapping (“/survey”) in survey form () method. I also created a wrapper class Survey form Dto for using the list of questions in thyme leaf. I also used for each loop from Thyme leaf to get the questions list. In survey html page I added the radio buttons for getting answers from the participants. Then I added the submit button to collect all the responses.
2. Then I refactored to breakdown my codes into smaller chunks and created a method save survey details () to save all the entries into H2 database. I put these codes in the method save survey details () and Post mapping is done with (“/save survey details”).
3. In the controller show thankyou form () method is added with Get mapping (“/thank you”) to navigate to the third page thankyou.html.
4. This page is with html and Thyme leaf codes along with an image added to it. Hyperlink is also added to navigate to the fourth page result.html.
5. Fourth page is the result page that uses Google chart API.
6. I was planning to bring the data from the database with questions and answers. So, get all survey details () method is added in the survey impl class.
7. Then started doing the coding in controller and modified the coding part several times to bring the connection to the result page that displays the survey response in column and pie chart.
8. I have the used the following constructs in my source code.

* th : each in thyme leaf
* H2 Database connection
* Class
* Methods
* Variables
* Collections
* Auto Id generator
* Local Date Time class
* Date Time Formatter class
* Collector class
* Grouping by () method
* Stream () method
* Various annotations

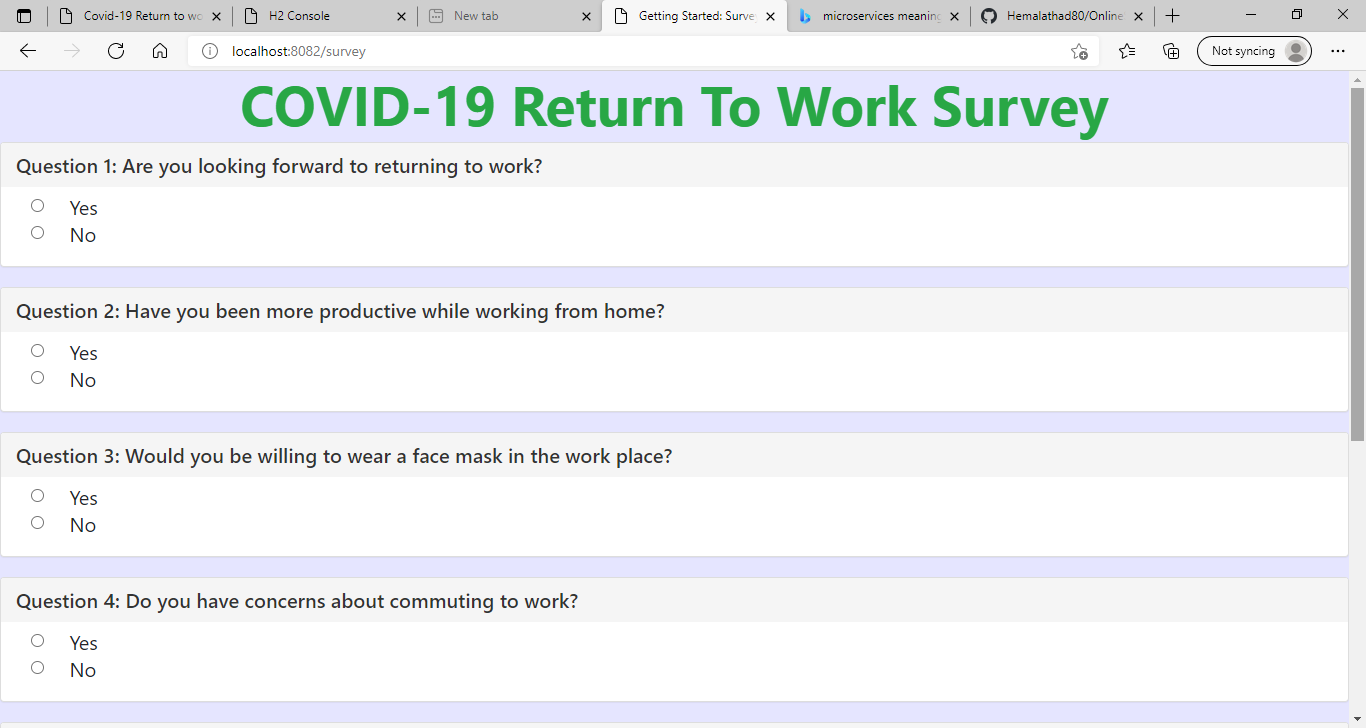
**How to run The Online Survey System Microservice**

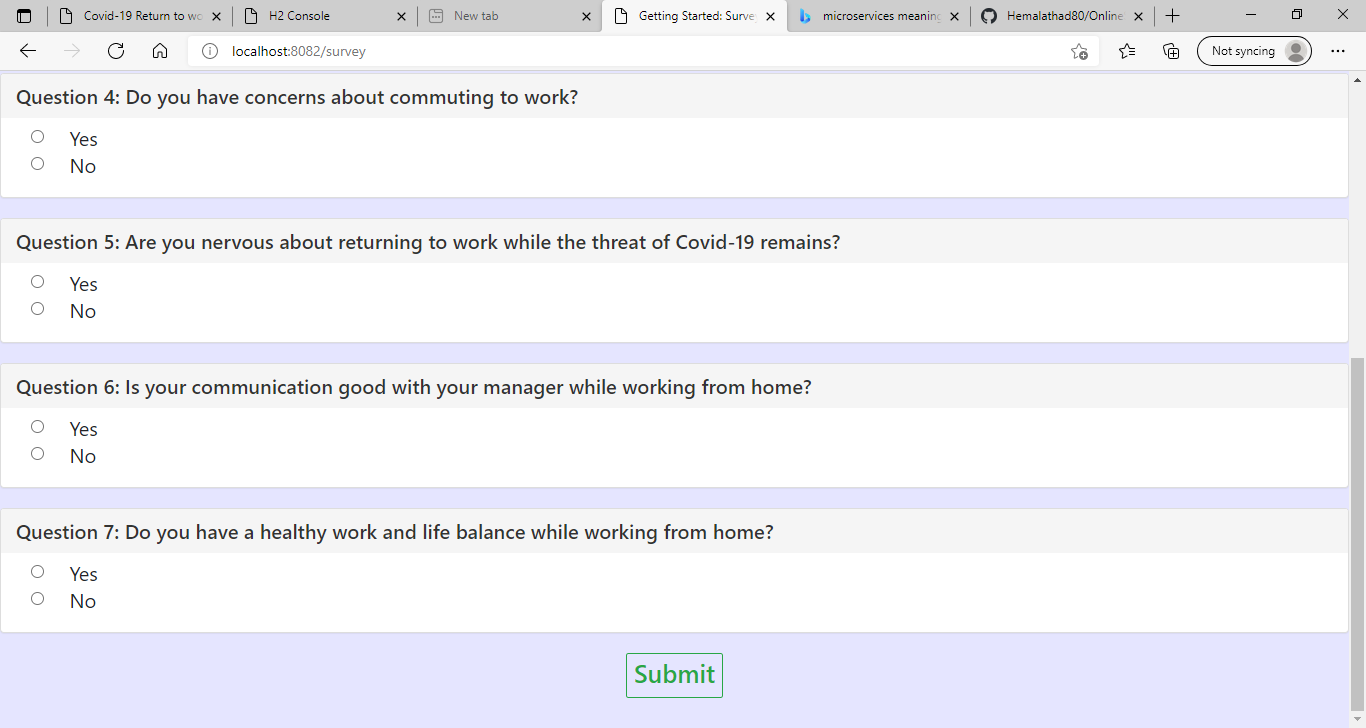
**STEP by step procedure**

1. Start running the application in IntelliJ J.
2. Type the URL <http://localhost:8082>/ in the web browser.
3. Index.html page gets displayed like below.

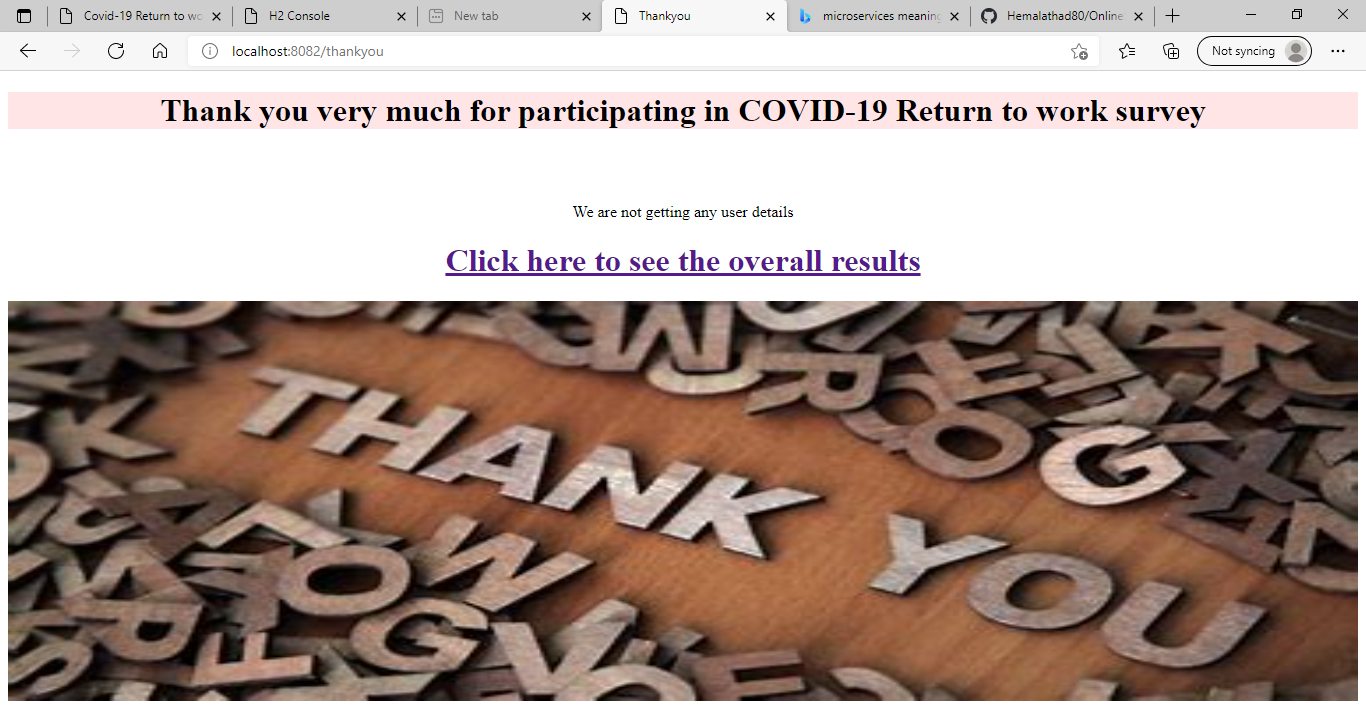


1. Click the hyperlink – “Please click here to take part in this survey” to navigate to the next page survey.html.
2. Below is the screenshot for the page.





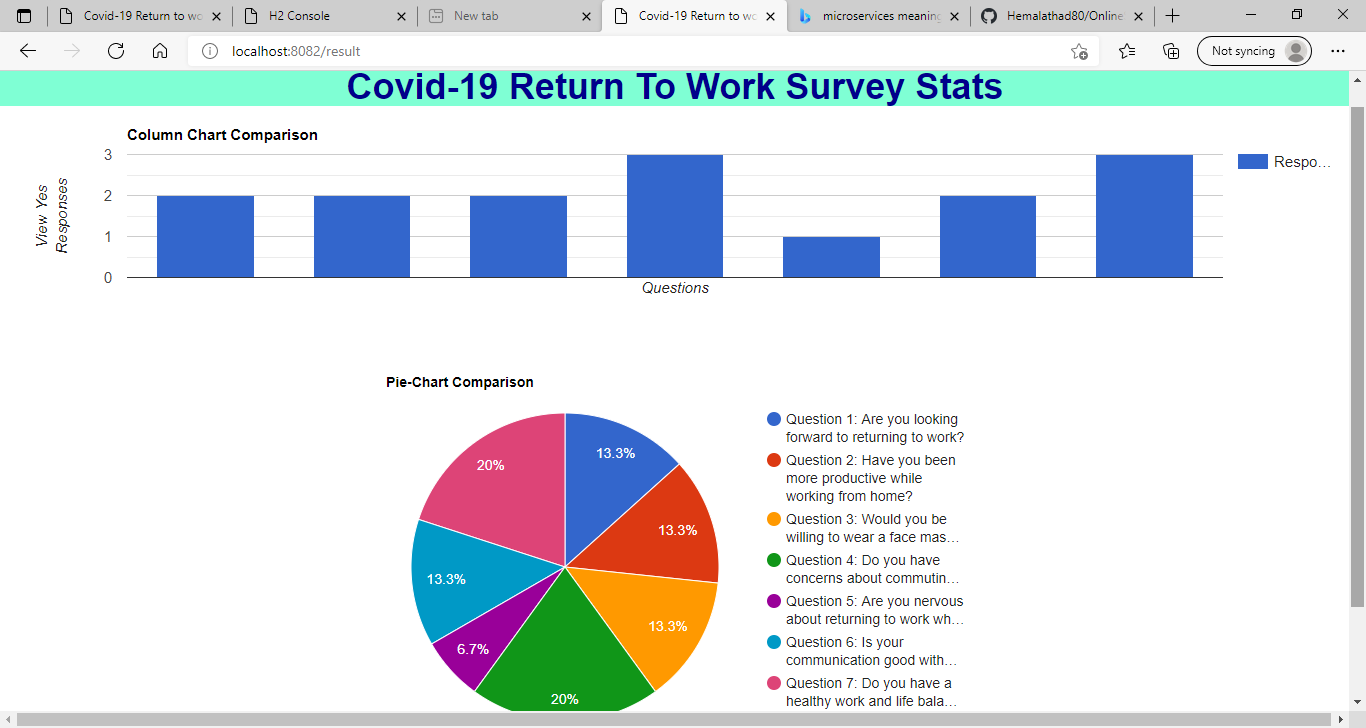
1. Please take part in this survey by answering yes or no questions by selecting the radio buttons.
2. Then press submit to navigate to the next page and also your responses will be stored in the database with current date and time.
3. Below is the screenshot of the third page thankyou.html.



1. Click the hyperlink – “Click here to see the overall results” navigates to last page result.html

Which takes the data from the database and shows the results in Google column chart and pie chart.

Below is the screen shot of the result page.



Important information

**H2 database log in details:**

**Driver class:** org.h2.Driver

**JDBC URL:** jdbc:h2:file:./data/survey

**User Name:** sa

**Password:** (empty)

**JUNIT TESTING**

Two test classes are created. They are Survey Controller Test class and Survey impl Test class.

For the Survey Controller Test class, the following test cases are built.

Show Welcome Page Returns Ok ()

Show Survey Page Returns Ok ()

Show thank you Page Returns Ok ()

Show result Page Returns Ok ()

For the Survey impl Test class, the following test case is built.

Find All Success ()

For the Survey Application Test class, the following test case is built.

context Loads ()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*