# Data Collection Form:

###### Project Leader: Ronak Makwana(18BCP093)

###### Group Members: Samyak Vora(18BCP098) & Shubham Rawat(18BCP110).

# About the Project:

Data collection Forms are the Forms that are used at the individual and cooperate level to collect data from the public. Here we have tried to create such a Data Collection Form which Fully Dynamic and can generate different types of questions according to users requirements. After that he can able to create a new static form which he can share to collect data.

# Overview of the Project:

* Data collection Form is a small project that helps users to make custom forms as per their varying requirements.
* This project is full stack developed.
* This project uses java swing for front-end GUI which helps user design the form as per the users necessity.
* XAMPP Apache server is used and XAMPP SQL services are used for back-end purpose.
* The responses can also be checked in the database in this Data Collection Form project.

# Abstract:

* Our team has developed this project to help those who want an application for daily/annual feedback or want to conduct quick surveys and maybe even conduct quizzes online.
* This app helps you create a form which is used to fill in different details by others based on your requirements.
* After generating the form, you can then share it with the targeted audience to fill in the required details.
* After the details have been filled the data received from the audience can be easily and efficiently extracted into database.
* This extracted data is stored in database and can be made easily available to you
* This app is also known as the Data Collection Form (DCF).
* The UI of our app is user friendly and can be easily accessed by anyone with some general knowledge about the internet.
* One of the advantages of using our app is that there is not only one way to make use of this app, but it can be used in many ways imaginable by the user.
* The data collected is also stored in our database for backup.

# Implementations:

* We used Java Swing to develop the front-end GUI.
* The following Swing components have been used in our Java Swing GUI:

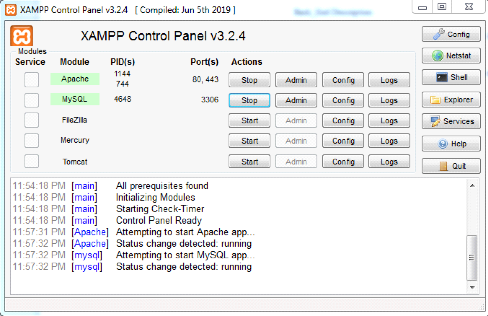
1. JTextfield
2. JTextArea
3. JCheckBox
4. JPasswordField
5. JComboBox
6. JRadioButton
7. JFrame
8. JPanel
9. JButton
10. ButtonGroup
11. ScrollPane

* We have created two frames so that one is used to make a form and another can be used to share the created form with the other users.
* For the requirement of server, we have used a XAMPP server, which is an open source cross platform web server solution pack.
* For transaction with the XAMPPserver and our project java files we have used php scripts.
* Inside the php scripts for transaction with the database we have created a Short Language Query from the data received, which when fired does its appropriate task with the database provided by the XAMPP server.

# Brief About the Project:

## Back-End Description:

* + For handling back-end requests we need a server. To fulfil that requirement, we have used a XAMPP server which is an open source cross platform tool.
  + From the XAMPP control panel we need to enable the Apache (For server) and the MySQL(For database) services for our project.

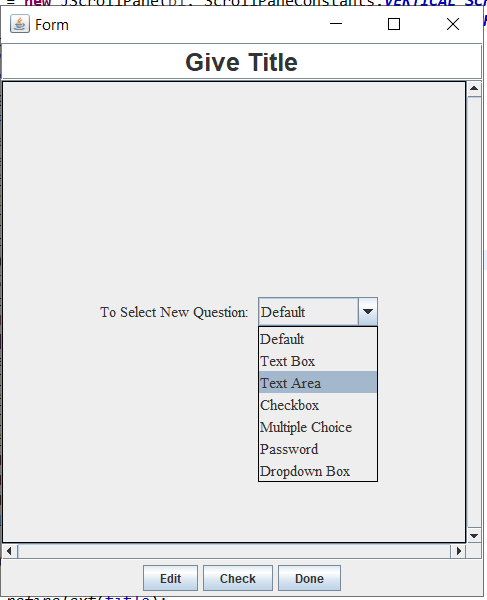


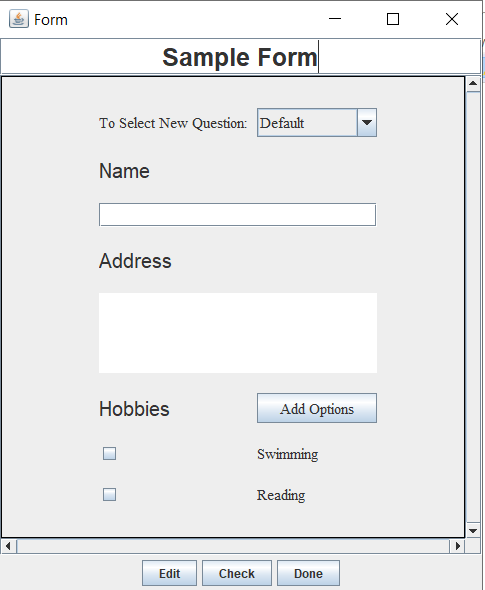
* + Now as the server is enabled all the files placed in "..\xampp\htdocs" can be executed.
  + A screenshot of a social media post

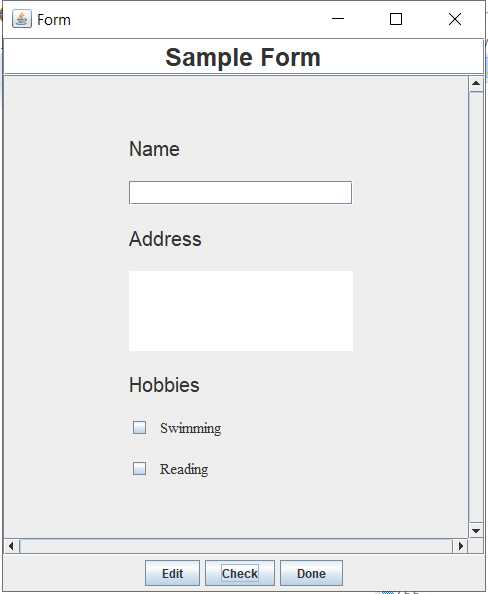
    Description automatically generatedFor e.g making connection to the XAMPP local server the file connect.php(provided in project files) is used which gives the output on the browser screen.
  + We will use this file every-time we need to establish connection with server.
  + For creating table and inserting into data we will get data from the form.
  + The form will create a post request to the php file (i.e now a website as reason of Apache services) and will post the data.
  + Now the php file will extract the data in required format and generate an appropriate SQL query to create a table or insert in a table.
  + This file name selection would be done appropriately from the java files.
  + For SQL query execution we will use the method in php mysqli\_query(connection,query);
  + After the query is executed the results wil be reflected in the database which can be viewed from the link provided in the UI portion of the form.

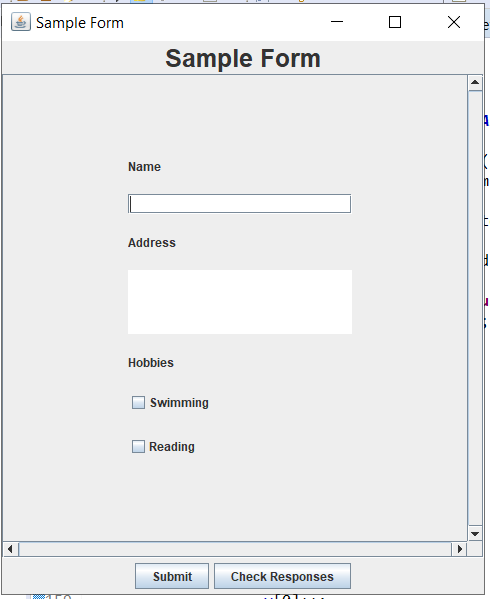
## Front-End Description:

* First we will create a new form where you can select any option from the drop-down list to create a new question of your choice.
* Run the program and a JFrame containing a drop-down box will appear, select the required questions from that. See the image below showing the Frame



* Now after selecting the required questions the form will appear like this where we will select:
* 1 JTextField - Name
* 1 JTextArea - Address
* 1 JCheckbox - Hobbies
* After that if you click on ‘Check’ button then you can look the user’s view of the form.



* After Checking the form Click on the ‘Done” button to create a new Frame containing the static form according to your requirement.
* This form can used to get data and store in the database.
* When you will click on the submit button, the data will be stored a table created in the Xampp database in your localhost which is explaoned earlier.
* You can check the data stored in the database by clicking ‘Check Response Button’.

## Advantages:

* A custom form can be created as per the user requirements.
* Project uses XAMPP server’s database services and not third party database services like Firebase from Google.
* For database management SQL is used which is highly efficient.
* Java applets are used which are cross platform i.e along with windows they can also be executed on Linux and OSI systems.

## Disadvantages:

* Java applets used in project were **outdated** long agoand do not support with latest jdk versions.

## Conclusion:

* Data collection form is a small proto-type of the implementation of custom forms (like Google forms).
* In this project we have implemented the form services with frontend and backend services (full stack).

## A few points to note about project:

* Authentication is not implemented.
* As authentication is not enabled anyone can access the database without any restrictions.