**PROFESSIONAL TRAINING REPORT**

# At

**Sathyabama Institute of Science and Technology (Deemed to be University)**

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering in Computer Science and Engineering BY

DASARI SRIKANTH

# (Reg. No. 38110120)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING SCHOOL OF COMPUTING**

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)**

**Accredited with Grade “A” by NAAC JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHEENNAI – 600 119**

## AUGUST 2020

**SATHYABAMA**

## INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)

**Accredited with Grade “A” by NAAC (Established under Section 3 of UGC Act, 1956)**

**JEPPIAAR NAGAR, RAJIV GANDHI SHALAI, CHENNAI – 600 119**

[**www.sathyabama.ac.in**](http://www.sathyabama.ac.in/)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**B ONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of DASARI SRIKANTH**(Reg. No. 38110120)** who carried out the project entitled “**BANK REGISTERATION SYSTEM**” under our supervision from APRIL 2020 to JUNE 2020

## Internal Guide

**Dr.A.Mary posonia M.E.,Ph.d.,**

**Head of the Department**

**Dr.S.VIGNESHWARI,M.E.,Ph.d.,**

**Submitted for Vivo voce Examination held on**

Internal Examiner External Examiner

# DECLARATION

I DASARI SRIKANTH (Reg. No.38110120)hereby declare that the Project Report entitled “**BANK REGISTRATION SYSTEM**” done by me under the guidance of **Dr.A.Mary posonia**  at **COURSERA ONLINE** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering

# DATE:

**PLACE: SIGNATURE OF THE CANDIDATE**

**ACKNOWLEDGEMENT**

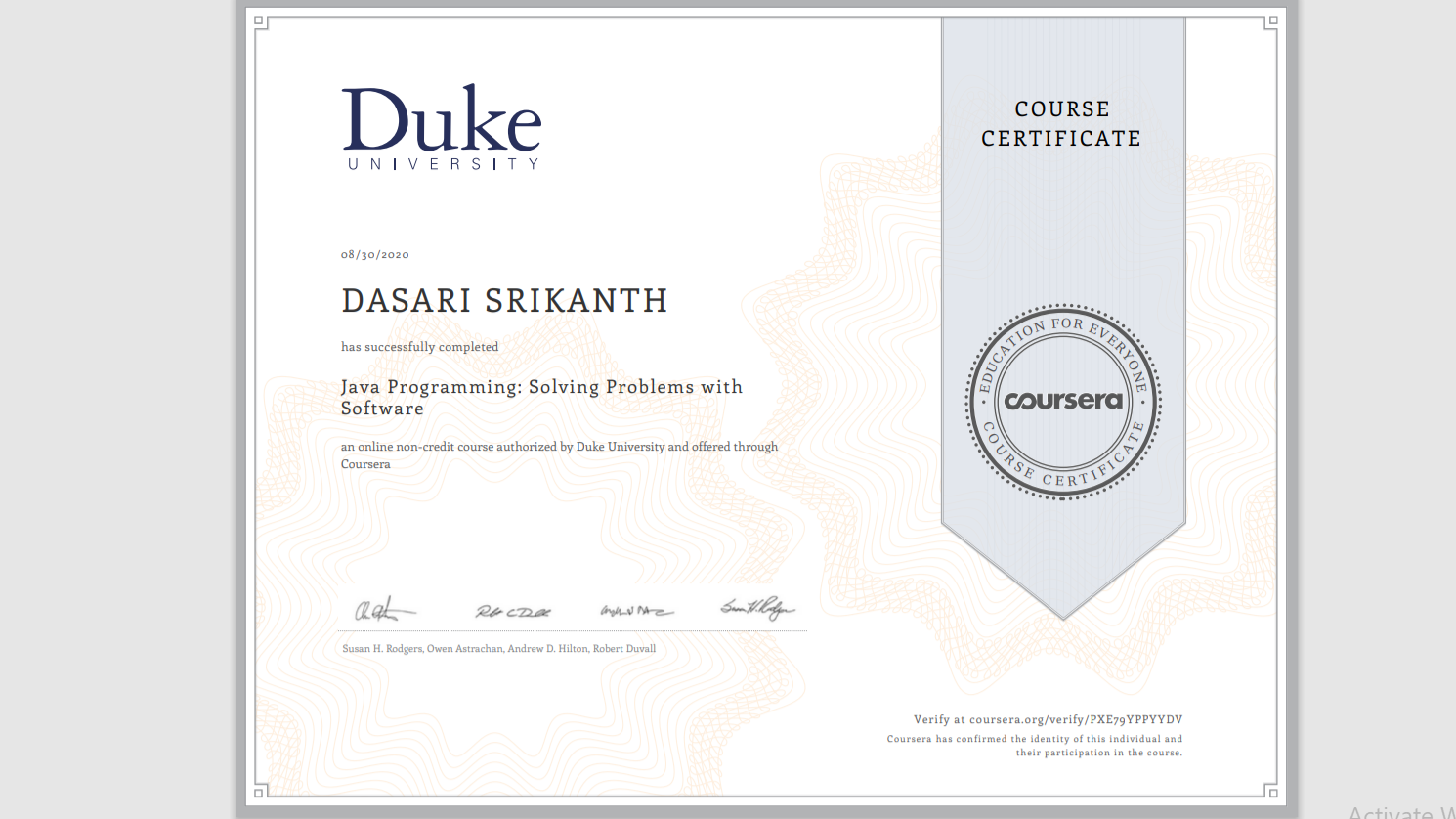
I am pleased to acknowledge my sincere thanks to **Board of management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA M.E., Ph.D., Dean, School of Computing** and **Dr. S. VIGNESHWARI M.E., Ph.D., and Dr. L. Lakshmana M.E., Ph.D., Heads of the Department, Department of Computer Science and Engineering** for providing me the necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

# TRAINING CERTIFICATE



**ABSTRACT:**

This project is a web based application specially for banking sector. This application will be used by bank administrator and bank employee only. by using this application a bank employee can open or edit accounts (saving, fixed, and recurring) for costumers.

Any transaction like money deposit, withdrawal, money transfer, demand draft facilities are available by using this web application. The interest amount is automatically updated to saving deposit or fixed deposit or recurring deposit.

Loan and insurance facility is also available in this web application. The administrator has total control over the web application. If the administrator grants the activities done by the employee then it is successfully done for the costumers.

The Bank Account Management System is an application for maintaining a person's account in a bank. In this project I tried to show the working of a banking account system and cover the basic functionality of a Bank Account Management System. To develop a project for solving financial applications of a customer in banking environment in order to nurture the needs of an end banking user by providing various ways to perform banking tasks. Also to enable the user’s work space to have additional functionalities which are not provided under a conventional banking project. The Bank Account Management System undertaken as a project is based on relevant technologies. The main aim of this project is to develop software for Bank Account Management System. This project has been developed to carry out the processes easily and quickly, which is not possible with the manuals systems, which are overcome by this software. This project is developed using PHP, HTML language and MYSQL use for database connection. Creating and managing requirements is a challenge of IT, systems and product development projects or indeed for any activity where you have to manage a contractual relationship. Organization need to effectively define and manage requirements to ensure they are meeting needs of the customer, while proving compliance and staying on the schedule and within budget. The impact of a poorly expressed requirement can bring a business out of compliance or even cause injury or death. Requirements definition and management is an activity that can deliver a high, fast return on investment. The project analyzes the system requirements and then comes up with the requirements specifications. It studies other related systems and then come up with system specifications. The system is then designed in accordance with specifications to satisfy the requirements. The system design is then implemented with MYSQL, PHP and HTML. The system is designed as an interactive and content management system. The content management system deals with data entry, validation confirm and updating whiles the interactive system deals with system interaction with the administration and users. Thus, above features of this project will save transaction time and therefore increase the efficiency of the system.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTERNO** | **TOPIC** | **PAGENO** |
| **1.** | **1.1 INTRODUCTION** | **8** |
| **2.** | **PROJECT SCOPE**  **2.1** Proposed System | **9-13** |
| **3.** | **SOFTWARE REQUIREMENTS**  **3.1** Installation Of Software | **14-17** |
| **4.** | **LANGUAGES LEARNT**  **4.1 Java**  **4.2 MySQL** | **18-27**  **28-33** |
| **5.** | **CODE**  **5.1 Html Code**  **5.2 Java Code** | **34-38**  **39-42** |
| **6.** | **OUTPUT** | **43-46** |
| **7.** | **CONCULSION** | **47** |
| **8.** | **REFERENCES** | **47** |

**1.1 INTRODUCTION:**

The Bank Account Management System is an application for maintaining a person's account in a bank. In this project I tried to show the working of a banking account system and cover the basic functionality of a Bank Account Management System. To develop a project for solving financial applications of a customer in banking environment in order to nurture the needs of an end banking user by providing various ways to perform banking tasks. Also to enable the user’s work space to have additional functionalities which are not provided under a conventional banking project. The Bank Account Management System undertaken as a project is based on relevant technologies. The main aim of this project is to develop software for Bank Account Management System. This project has been developed to carry out the processes easily and quickly, which is not possible with the manuals systems, which are overcome by this software. This project is developed using PHP, HTML language and MYSQL use for database connection. Creating and managing requirements is a challenge of IT, systems and product development projects or indeed for any activity where you have to manage a contractual relationship. Organization need to effectively define and manage requirements to ensure they are meeting needs of the customer, while proving compliance and staying on the schedule and within budget. The impact of a poorly expressed requirement can bring a business out of compliance or even cause injury or death. Requirements definition and management is an activity that can deliver a high, fast return on investment. The project analyzes the system requirements and then comes up with the requirements specifications. It studies other related systems and then come up with system specifications. The system is then designed in accordance with specifications to satisfy the requirements. The system design is then implemented with MYSQL, PHP and HTML. The system is designed as an interactive and content management system. The content management system deals with data entry, validation confirm and updating whiles the interactive system deals with system interaction with the administration and users. Thus, above features of this project will save transaction time and therefore increase the efficiency of the system.

**2.1 PROPOSED SYSTEM:**

The Bank Account Management System is an application for maintaining a person's account in a bank. In this project I tried to show the working of a banking account system and cover the basic functionality of a Bank Account Management System. To develop a project for solving financial applications of a customer in banking environment in order to nurture the needs of an end banking user by providing various ways to perform banking tasks. Also to enable the user’s work space to have additional functionalities which are not provided under a conventional banking project. The Bank Account Management System undertaken as a project is based on relevant technologies. The main aim of this project is to develop software for Bank Account Management System. This project has been developed to carry out the processes easily and quickly, which is not possible with the manuals systems, which are overcome by this software. This project is developed using PHP, HTML language and MYSQL use for database connection. Creating and managing requirements is a challenge of IT, systems and product development projects or indeed for any activity where you have to manage a contractual relationship. Organization need to effectively define and manage requirements to ensure they are meeting needs of the customer, while proving compliance and staying on the schedule and within budget. The impact of a poorly expressed requirement can bring a business out of compliance or even cause injury or death. Requirements definition and management is an activity that can deliver a high, fast return on investment. The project analyzes the system requirements and then comes up with the requirements specifications. It studies other related systems and then come up with system specifications. The system is then designed in accordance with specifications to satisfy the requirements. The system design is then implemented with MYSQL, PHP and HTML. The system is designed as an interactive and content management system. The content management system deals with data entry, validation confirm and updating whiles the interactive system deals with system interaction with the administration and users. Thus, above features of this project will save transaction time and therefore increase the efficiency of the system.

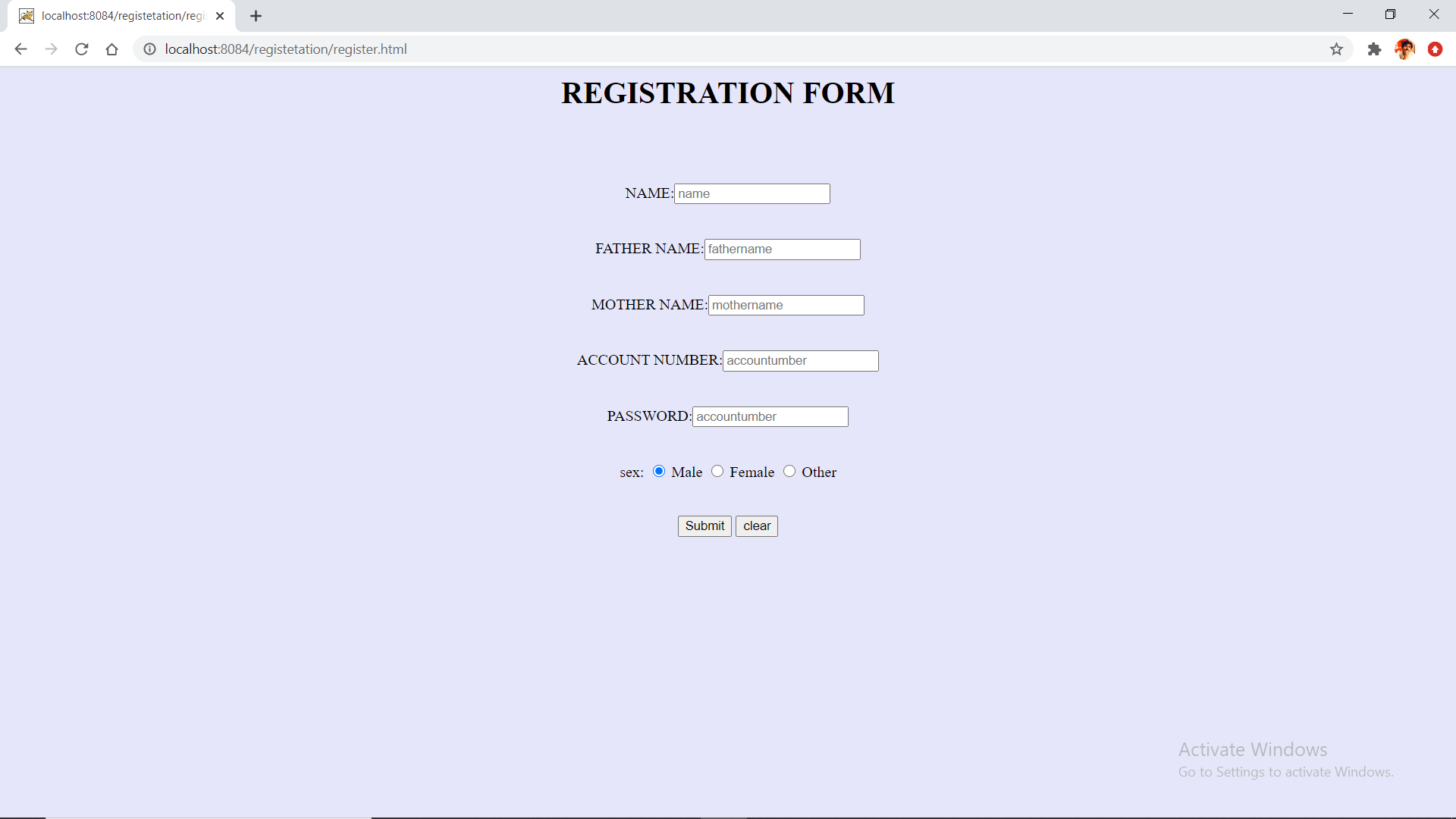
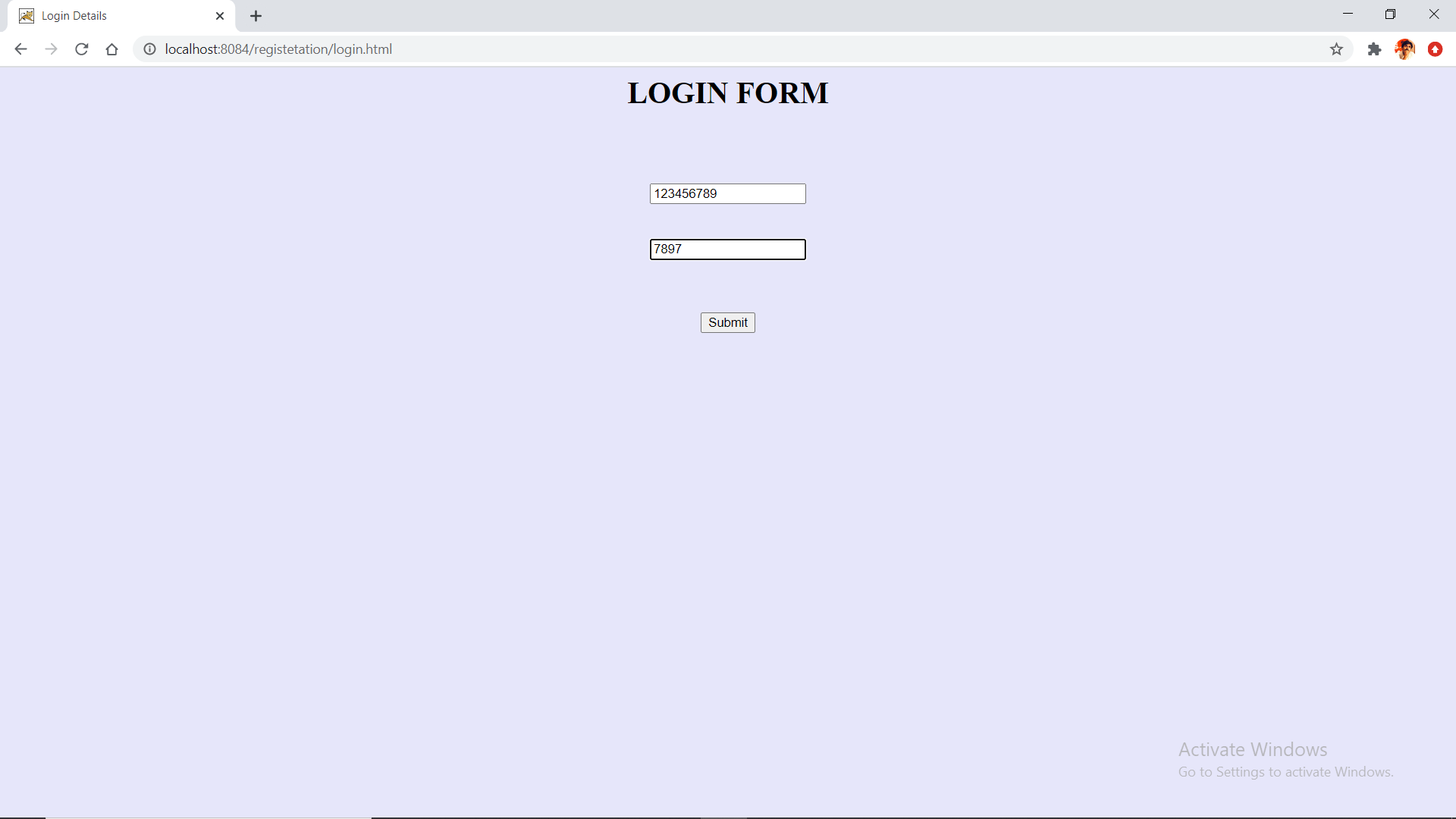
This project is a web based application specially for banking sector. This application will be used by bank administrator and bank employee only. by using this application a bank employee can open or edit accounts (saving, fixed, and recurring) for costumers.

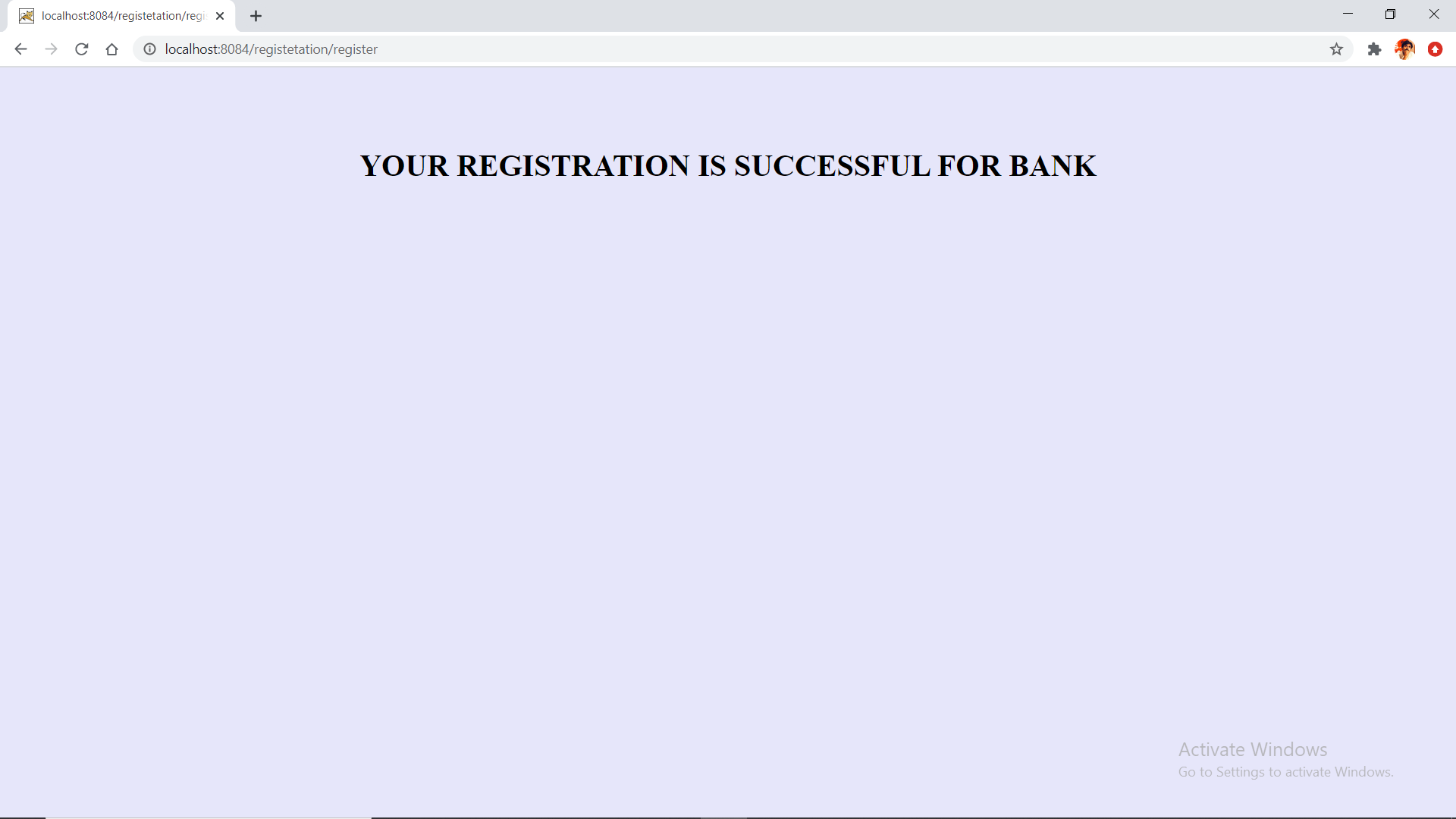
Any transaction like money deposit, withdrawal, money transfer, demand draft facilities are available by using this web application. The interest amount is automatically updated to saving deposit or fixed deposit or recurring deposit.

Loan and insurance facility is also available in this web application. The administrator has total control over the web application. If the administrator grants the activities done by the employee then it is successfully done for the costumers.

**PROPOSED SYSTEM MODEL:**

****

**** ****

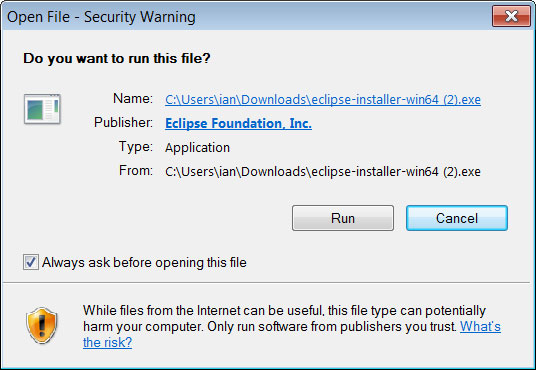
****

**3.1 INSTALLATION OF SOFTWARE(ECLIPSE)**

**1) Download the Eclipse installer:**

[**https://eclipse.org/downloads**](https://eclipse.org/downloads)

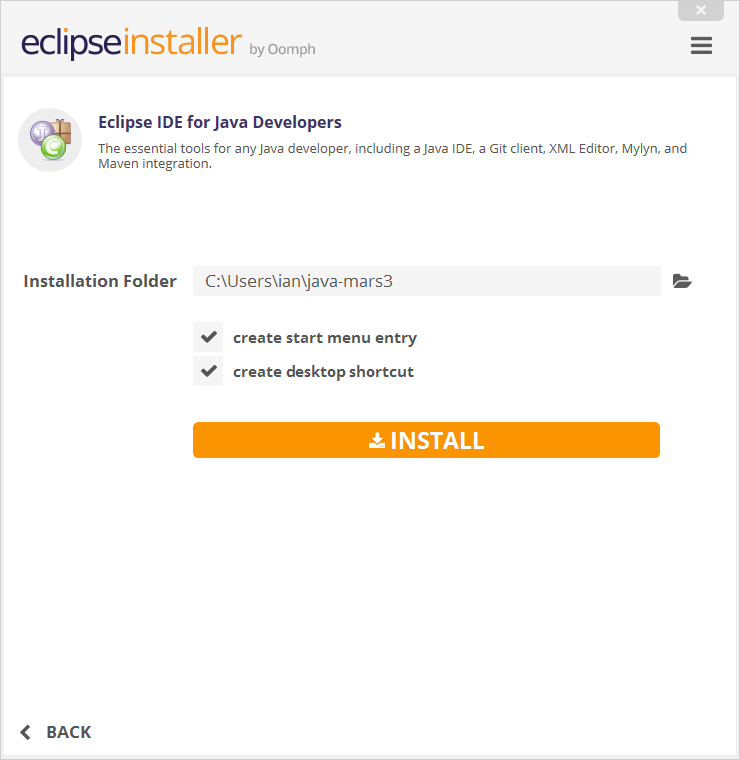
**2) Start The Eclipse installer Executable:**

****

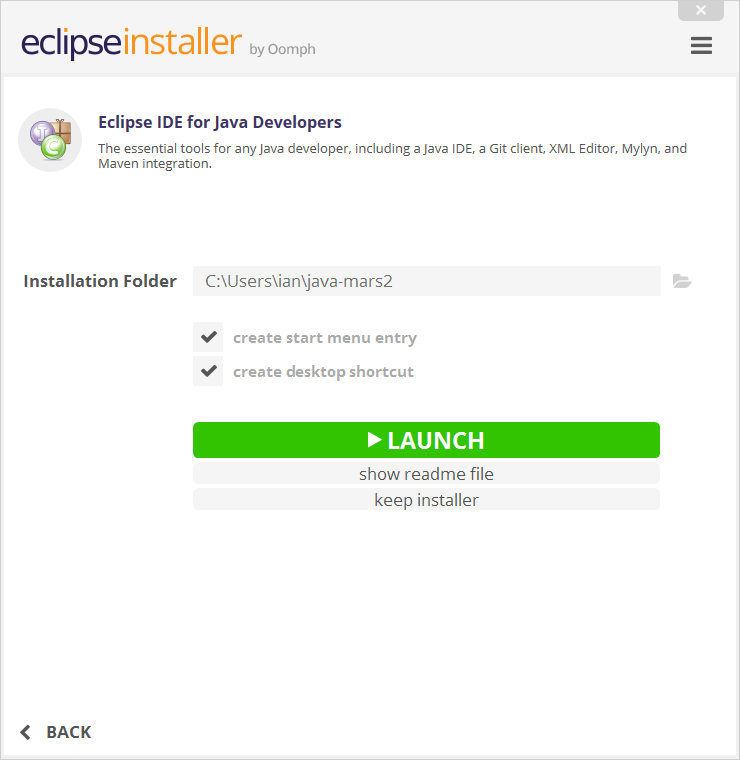
**3) Select The Package to Install:**

****

**4) Select the Installation Folder:**

****

**5) Launch Eclipse:**

****

**LANGUAGES LEARNT:**

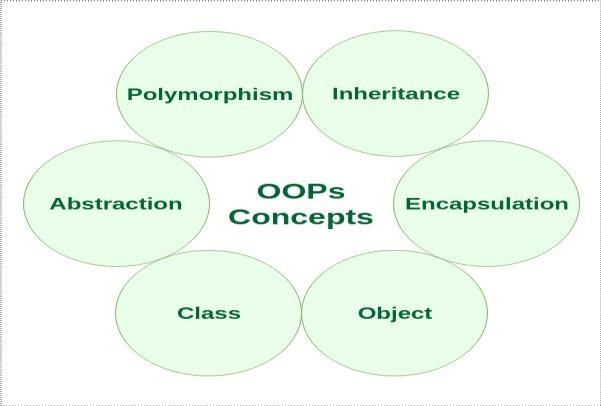
**4.1)JAVA:**

Java is a [general-purpose](https://en.wikipedia.org/wiki/General-purpose_language) [programming language](https://en.wikipedia.org/wiki/Programming_language) that is [class-based](https://en.wikipedia.org/wiki/Class-based_programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), and designed to have as few implementation [dependencies](https://en.wikipedia.org/wiki/Dependency_(computer_science)) as possible. It is intended to let [application developers](https://en.wikipedia.org/wiki/Application_developer) write once, run anywhere (WORA),meaning that [compiled](https://en.wikipedia.org/wiki/Compiler) Java code can run on all platforms that support Java without the need for recompilation.[[18]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-design_goals-18) Java applications are typically compiled to [bytecode](https://en.wikipedia.org/wiki/Java_bytecode) that can run on any [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) regardless of the underlying [computer architecture.](https://en.wikipedia.org/wiki/Computer_architecture) The [syntax](https://en.wikipedia.org/wiki/Syntax_(programming_languages)) of [Java](https://en.wikipedia.org/wiki/Java_(software_platform)) is similar to [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B), but it has fewer [low-level](https://en.wikipedia.org/wiki/Low-level_programming_language) facilities than either of them. As of 2019, Java was one of the most [popular programming languages in use](https://en.wikipedia.org/wiki/Measuring_programming_language_popularity) according to [GitHub](https://en.wikipedia.org/wiki/GitHub), particularly for [client-](https://en.wikipedia.org/wiki/Client%E2%80%93server) [server](https://en.wikipedia.org/wiki/Client%E2%80%93server) [web applications](https://en.wikipedia.org/wiki/Web_applications), with a reported 9 million developers.

OOPS Concepts

Object means a real-world entity such as a pen, chair, table, computer, watch, etc. Object- Oriented Programming is a methodology or paradigm to design a program using classes and objects. It simplifies software development and maintenance by providing some concepts:

* [Object](https://www.javatpoint.com/object-and-class-in-java)
* Class
* [Inheritance](https://www.javatpoint.com/inheritance-in-java)
* [Polymorphism](https://www.javatpoint.com/runtime-polymorphism-in-java)
* [Abstraction](https://www.javatpoint.com/abstract-class-in-java)
* [Encapsulation](https://www.javatpoint.com/encapsulation)



Object: is a bundle of data and its behavior (often known as methods).

Class: A class can be considered as a blueprint using which you can create as many objects as you like. For example, here we have a class Website that has two data members (also known as fields, instance variables and object states). This is just a blueprint, it does not represent any website, however using this we can create Website objects (or instances) that represents the websites. We have created two objects, while creating objects we provided separate properties to the objects using constructor.

[Constructor](https://beginnersbook.com/2013/03/constructors-in-java/) looks like a method but it is in fact not a method. It’s name is same as class name and it does not return any value. You must have seen this statement in almost all the programs I have shared above:

Abstraction: Abstraction is a process where you show only “relevant” data and “hide” unnecessary details of an object from the user. For example, when you login to your bank account online, you enter your user\_id and password and press login, what happens when you press login, how the input data sent to server, how it gets verified is all abstracted away from the you.

Encapsulation: Encapsulation simply means binding object state(fields) and behavior(methods) together. If you are creating class, you are doing encapsulation.

Inheritance: The process by which one class acquires the properties and functionalities of another class is calle[d inheritance](https://beginnersbook.com/2013/03/inheritance-in-java/). Inheritance provides the idea of reusability of code and each sub class defines only those features that are unique to it, rest of the features can be inherited from the parent class.

1. Inheritance is a process of defining a new class based on an existing class by extending its common data members and methods.
2. Inheritance allows us to reuse of code, it improves reusability in your java application.
3. The parent class is called the base class or super class. The child class that extends the base class is called the derived class or sub class or child class.

[Types of Inheritance](https://beginnersbook.com/2013/05/java-inheritance-types/):

Single Inheritance: refers to a child and parent class relationship where a class extends the another class.

Multilevel inheritance: refers to a child and parent class relationship where a class extends the child class. For example class A extends class B and class B extends class C.

Hierarchical inheritance: refers to a child and parent class relationship where more than one classes extends the same class. For example, class B extends class A and class C extends class A.

Multiple Inheritance: refers to the concept of one class extending more than one classes, which means a child class has two parent classes. Java doesn’t support multiple inheritance, read more about it

[**Polymorphism**](https://beginnersbook.com/2013/03/polymorphism-in-java/) is a object oriented programming feature that allows us to perform a single action in different ways. For example, lets say we have a class Animal that has a method animalSound(), here we cannot give implementation to this method as we do not know which Animal class would extend Animal class. So, we make this method abstract like this:

### [Types of Polymorphism](https://beginnersbook.com/2013/04/runtime-compile-time-polymorphism/)

1. Static Polymorphism
2. Dynamic Polymorphism

**Static Polymorphism:** Polymorphism that is resolved during compiler time is known static

Polymorphism.

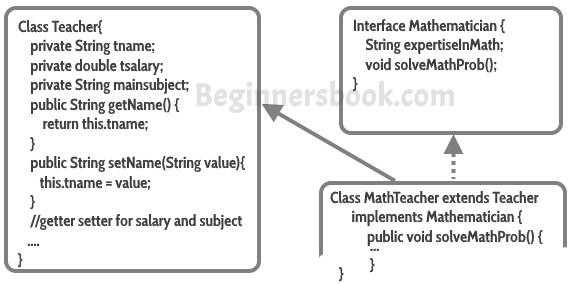
**Dynamic Polymorphism**

It is also known as Dynamic Method Dispatch. Dynamic polymorphism is a process in which a call to an overridden method is resolved at runtime rather, thats why it is called runtime polymorphism.

Abstract Class and methods in OOPs Concepts Abstract method:

1. A method that is declared but not defined. Only method signature no body.
2. Declared using the abstract keyword.

An interface is a blueprint of a class, which can be declared by using interface keyword. Interfaces can contain only constants and abstract methods (methods with only signatures no body). Like abstract classes, Interfaces cannot be instantiated, they can only be implemented by classes or extended by other interfaces. Interface is a common way to achieve full abstraction in Java.



Access Specifiers:

Well, you must have seen public, private keyword in the examples I have shared above. They are calle[d access specifiers](https://beginnersbook.com/2013/05/java-access-modifiers/) as they decide the scope of a data member, method or class.

There are four types of access specifiers in java:

public: Accessible to all. Other objects can also access this member variable or function.

private: Not accessible by other objects. Private members can be accessed only by the methods in the same class. Object accessible only in class in which they are declared.

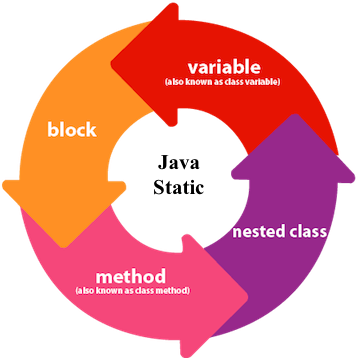
**protected:** The scope of a protected variable is within the class which declares it and in the class which inherits from the class (Scope is class and subclass).

**Default:** Scope is Package Level. We do not need to explicitly mention default as when we do not mention any access specifier it is considered as default.

### Static Key Word

The **static keyword** in [Java](https://www.javatpoint.com/java-tutorial) is used for memory management mainly. We can apply static keyword with [variables,](https://www.javatpoint.com/java-variables) methods, blocks and [nested classes](https://www.javatpoint.com/java-inner-class). The static keyword belongs to the class than an instance of the class.The static can be:

* 1. Variable (also known as a class variable)
  2. Method (also known as a class method)
  3. Block
  4. Nested class



### Spring Frame Work

The Spring Framework provides a comprehensive programming and configuration model for modern Java-based enterprise applications - on any kind of deployment platform.A key element of Spring is infrastructural support at the application level: Spring focuses on the "plumbing" of enterprise applications so that teams can focus on application-level business logic, without unnecessary ties to specific deployment environments.

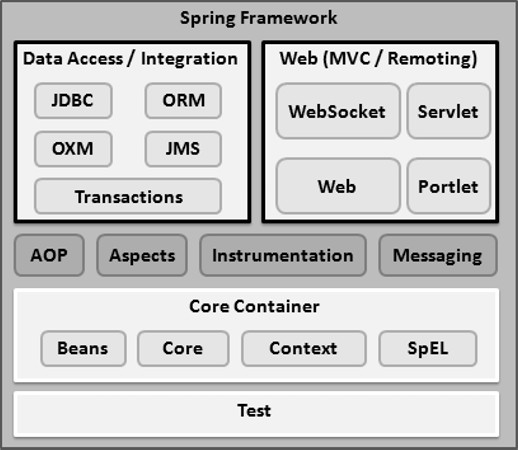
**Features** [:Core technologies](https://docs.spring.io/spring-framework/docs/current/spring-framework-reference/core.html): dependency injection, events, resources, i18n, validation, data binding, type conversion, SpEL, AOP.

[Testing](https://docs.spring.io/spring-framework/docs/current/spring-framework-reference/testing.html): mock objects, TestContext framework, Spring MVC Test, WebTestClient.

[Data Access](https://docs.spring.io/spring-framework/docs/current/spring-framework-reference/data-access.html): transactions, DAO support, JDBC, ORM, Marshalling XML. [Spring MVC](https://docs.spring.io/spring/docs/current/spring-framework-reference/web.html) and [Spring WebFlux](https://docs.spring.io/spring/docs/current/spring-framework-reference/web-reactive.html) web frameworks.

[Integration](https://docs.spring.io/spring-framework/docs/current/spring-framework-reference/integration.html): remoting, JMS, JCA, JMX, email, tasks, scheduling, cache.

[Languages](https://docs.spring.io/spring-framework/docs/current/spring-framework-reference/languages.html): Kotlin, Groovy, dynamic languages. The Spring framework is a layered architecture which consists of several modules. All modules are built on the top of its core container. These modules provide everything that a developer may need for use in the enterprise application development. He is always free to choose what features he needs and eliminate the modules which are of no use. It's modular architecture enables integration with other frameworks without much ha ssle.



### Data Access/Integration

The Data Access/Integration layer consists of the JDBC, ORM, OXM, JMS and Transaction modules whose detail is as follows

* + - The JDBC module provides a JDBC-abstraction layer that removes the need for tedious JDBC related coding.
    - The ORM module provides integration layers for popular object-relational mapping APIs, including JPA, JDO, Hibernate, and iBatis.

### Web

The Web layer consists of the Web, Web-MVC, Web-Socket, and Web-Portlet modules the details of which are as follows −

* + - The Web module provides basic web-oriented integration features such as multipart file-upload functionality and the initialization of the IoC container using servlet listeners and a web-oriented application context.
    - The Web-MVC module contains Spring's Model-View-Controller (MVC) implementation for web applications.
    - The Web-Socket module provides support for WebSocket-based, two-way communication between the client and the server in web applications
    - The Web-Portlet module provides the MVC implementation to be used in a portlet environment and mirrors the functionality of Web-Servlet module.

### Spring Boot

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. This chapter will give you an introduction to Spring Boot and familiarizes you with its basic concepts.

Micro Service is an architecture that allows the developers to develop and deploy services independently. Each service running has its own process and this achieves the lightweight model to support business applications.

### Advantages

Micro services offers the following advantages to its developers −

* Easy deployment
* Simple scalability
* Compatible with Containers
* Minimum configuration
* Lesser production time

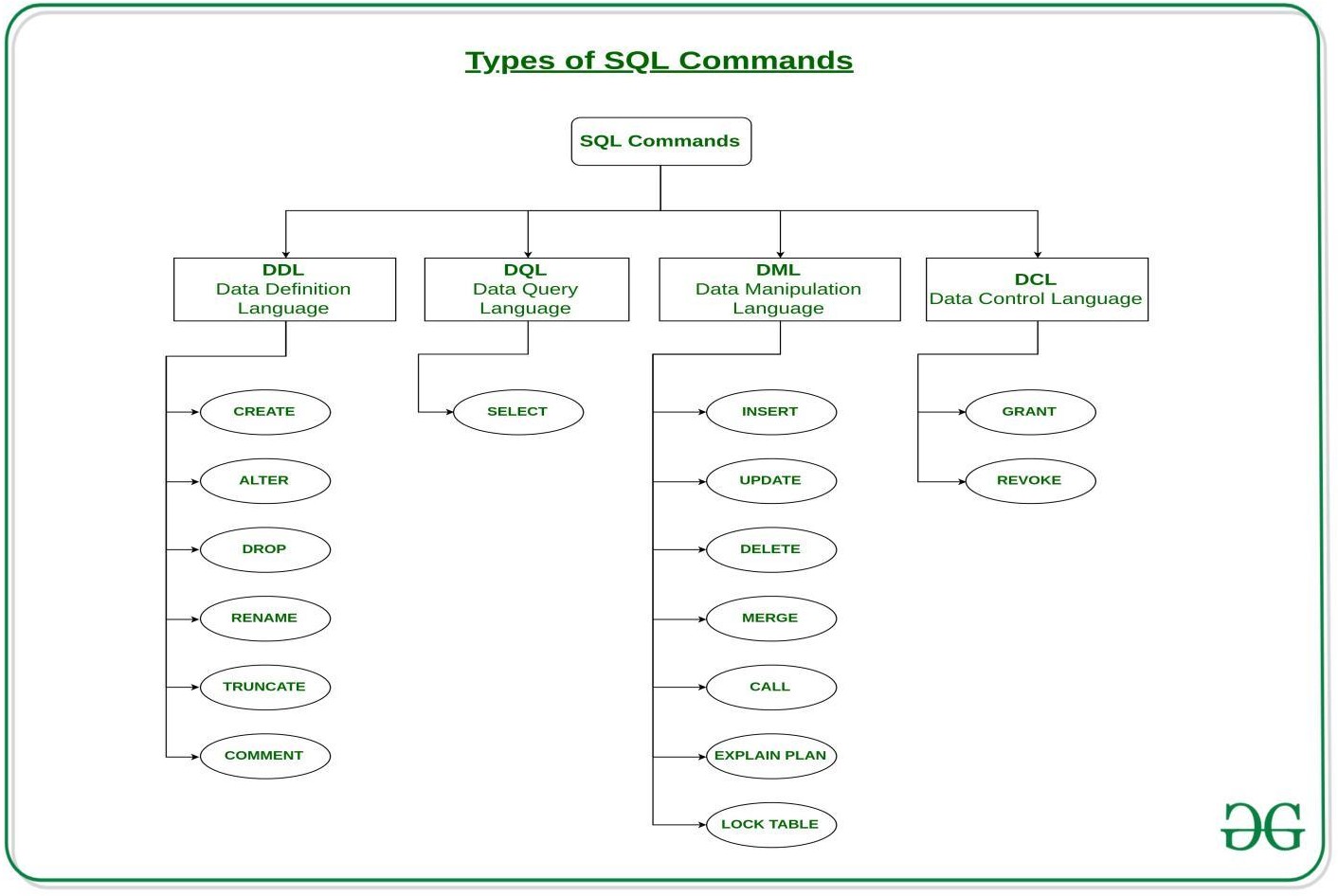
Spring Boot is designed with the following goals −

* To avoid complex XML configuration in Spring
* To develop a production ready Spring applications in an easier way
* To reduce the development time and run the application independently
* Offer an easier way of getting started with the application

**4.2)Mysql:**

## Data Base Management:

Database Management System (DBMS) is a software for storing and retrieving users' data while considering appropriate security measures. It consists of a group of programs which manipulate the database

We discussed four main types of databases: text databases, desktop database programs, relational database management systems (RDMS), and NoSQL and object-oriented databases. We also talked about two ways to categorize databases based on their logical design: operational databases and database warehouses.

[Primary Key](https://beginnersbook.com/2015/04/primary-key-in-dbms/) – A primary is a column or set of columns in a table that uniquely identifies tuples (rows) in that table.

[Super Key](https://beginnersbook.com/2015/04/super-key-in-dbms/) – A super key is a set of one of more columns (attributes) to uniquely identify rows in a table.

[Candidate Key](https://beginnersbook.com/2015/04/candidate-key-in-dbms/) – A super key with no redundant attribute is known as candidate key

[Alternate Key](https://beginnersbook.com/2015/04/alternate-key-in-dbms/) – Out of all candidate keys, only one gets selected as primary key, remaining keys are known as alternate or secondary keys.

[Composite Key](https://beginnersbook.com/2015/04/composite-key-in-dbms/) – A key that consists of more than one attribute to uniquely identify rows (also known as records & tuples) in a table is called composite key.

[Foreign Key](https://beginnersbook.com/2015/04/foreign-key-in-dbms/) – Foreign keys are the columns of a table that points to the primary key of another table. They act as a cross-reference between tables.

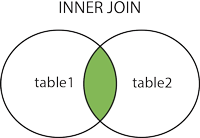
Joins

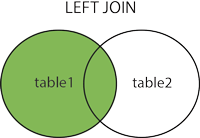
A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

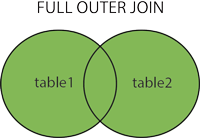
Let's look at a selection from the "Orders" table:

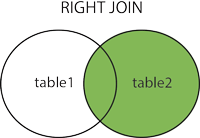
Here are the different types of the JOINs in SQL:

* + (INNER) JOIN: Returns records that have matching values in both tables
  + LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table
  + RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table
  + FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table









**Sub Queries in Sql:**

A subquery is a SQL query nested inside a larger query.

* + A subquery may occur in :
    - - A SELECT clause
    - - A FROM clause
    - - A WHERE clause
  + The subquery can be nested inside a SELECT, INSERT, UPDATE, or DELETE statement or inside another subquery.
  + A subquery is usually added within the WHERE Clause of another SQL SELECT statement.
  + You can use the comparison operators, such as >, <, or =. The comparison operator can also be a multiple-row operator, such as IN, ANY, or ALL.
  + A subquery is also called an inner query or inner select, while the statement containing a subquery is also called an outer query or outer select.
  + The inner query executes first before its parent query so that the results of an inner query can be passed to the outer query

**SQL Aggregate Functions:**

* AVG() – Returns the average value.
* COUNT() – Returns the number of rows.
* MAX() – Returns the largest value.
* MIN() – Returns the smallest value.
* SUM() – Returns the sum.

**5)CODE:**

**5.1)HTML CODE:  
 (index.html)**

**<!DOCTYPE html>**

**<!--**

**To change this license header, choose License Headers in Project Properties.**

**To change this template file, choose Tools | Templates**

**and open the template in the editor.**

**-->**

**<html>**

**<head>**

**<title>Bank Details For Register or Login</title>**

**<style>**

**body {**

**background-color: #E6E6FA;**

**}**

**h1**

**{**

**text-align: center;**

**}**

**form**

**{**

**text-align: center;**

**}**

**</style>**

**</head>**

**<body>**

**<br>**

**<br>**

**<br>**

**<h1>BANK REGISTRATION SYSTEM</h1><br><br><br>**

**<form>**

**<button>**

**<a href="register.html">Register</a></button>**

**<br>**

**<br><br><br>**

**<button>**

**<a href="login.html">Login</a>**

**</button>**

**</form>**

**</body>**

**</html>**

**2)register.html**

**<html>**

**<head>**

**<style>**

**body {**

**background-color: #E6E6FA;**

**}**

**form**

**{**

**text-align: center;**

**}**

**h1**

**{**

**text-align: center;**

**}**

**</style>**

**</head>**

**<body>**

**<h1>REGISTRATION FORM</h1><br><br><br>**

**<form action="register" method="post">**

**NAME:<input type="text" name="name" placeholder="name"><br>**

**<br>**

**<br>**

**FATHER NAME:<input type="text" name="fname" placeholder="fathername"><br>**

**<br>**

**<br>**

**MOTHER NAME:<input type="text" name="mname" placeholder="mothername"><br>**

**<br>**

**<br>**

**ACCOUNT NUMBER:<input type="number" name="ano" placeholder="accountumber"><br>**

**<br>**

**<br>**

**PASSWORD:<input type="password" name="pass" placeholder="accountumber"><br><br><br>**

**sex:**

**<input type="radio" name="gender" value="male" checked> Male**

**<input type="radio" name="gender" value="female"> Female**

**<input type="radio" name="gender" value="other"> Other<br>**

**<br>**

**<br>**

**<input type="submit" value="Submit">**

**<input type="button" value="clear">**

**<br>**

**<br>**

**</form>**

**</body>**

**</html>**

**3)login.html**

**<html>**

**<head>**

**<title>Login Details</title>**

**<style>**

**body {**

**background-color: #E6E6FA;**

**}**

**form**

**{**

**text-align: center;**

**}**

**h1**

**{**

**text-align: center;**

**}**

**</style>**

**</head>**

**<body>**

**<form action="login" method="post">**

**<h1>LOGIN FORM</h1><br><br><br>**

**<input type="text" name="accno" placeholder="Enter Account Numaber:"><br><br><br>**

**<input type="text" name="pass" placeholder="Enter Your Password"><br><br><br><br>**

**<input type="submit">**

**</form>**

**</body>**

**</html>**

**5.2)JAVA CODE:**

**1)register.java:**

**/\***

**\* To change this license header, choose License Headers in Project Properties.**

**\* To change this template file, choose Tools | Templates**

**\* and open the template in the editor.**

**\*/**

**import java.io.IOException;**

**import java.io.PrintWriter;**

**import java.sql.Connection;**

**import java.sql.DriverManager;**

**import java.sql.SQLException;**

**import javax.servlet.ServletException;**

**import javax.servlet.http.HttpServlet;**

**import javax.servlet.http.HttpServletRequest;**

**import javax.servlet.http.HttpServletResponse;**

**/\*\***

**\***

**\* @author mylaptop**

**\*/**

**public class register extends HttpServlet {**

**@Override**

**public void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException {**

**response.setContentType("text/html;charset=UTF-8");**

**PrintWriter out = response.getWriter();**

**try {**

**String name = request.getParameter("name");**

**String fname=request.getParameter("fname");**

**String mname=request.getParameter("mname");**

**String sex=request.getParameter("gender");**

**String accountno = request.getParameter("ano");**

**String pass=request.getParameter("pass");**

**Class.forName("com.mysql.jdbc.Driver");**

**Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/bharath","root","bharath");**

**con.createStatement().executeUpdate("insert into accountdetails values('"+name+"','"+fname+"','"+mname+"','"+sex+"','"+accountno+"','"+pass+"')");**

**out.println("<html><head> <style>h1{text-align: center;}body{ background-color: #E6E6FA;}</style></head><body><br><br><br><h1>YOUR REGISTRATION IS SUCCESSFUL FOR BANK</h1></html>");**

**}**

**catch (Exception e) {**

**out.println(e);**

**}**

**}**

**}**

**2)login.java:**

**/\***

**\* To change this license header, choose License Headers in Project Properties.**

**\* To change this template file, choose Tools | Templates**

**\* and open the template in the editor.**

**\*/**

**import java.io.IOException;**

**import java.io.PrintWriter;**

**import static java.lang.Class.forName;**

**import java.sql.Connection;**

**import java.sql.DriverManager;**

**import java.sql.ResultSet;**

**import java.sql.SQLException;**

**import java.sql.Statement;**

**import java.util.logging.Level;**

**import java.util.logging.Logger;**

**import javax.servlet.ServletException;**

**import javax.servlet.http.HttpServlet;**

**import javax.servlet.http.HttpServletRequest;**

**import javax.servlet.http.HttpServletResponse;**

**/\*\***

**\***

**\* @author mylaptop**

**\*/**

**public class login extends HttpServlet {**

**@Override**

**public void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException {**

**response.setContentType("text/html;charset=UTF-8");**

**PrintWriter out = response.getWriter();**

**String accountno=request.getParameter("accno");**

**String password=request.getParameter("pass");**

**int c=0;**

**try**

**{**

**Class.forName("com.mysql.jdbc.Driver");**

**Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/bharath","root","bharath");**

**Statement stmt=con.createStatement();**

**String sql="select \* from accountdetails";**

**stmt.executeQuery(sql);**

**ResultSet rs=stmt.executeQuery(sql);**

**while(rs.next())**

**{**

**if(accountno.equals(rs.getString(5)) && password.equals(rs.getString(6)))**

**{**

**//out.println("you are logged in");**

**out.println("<html><head> <style>h1{text-align: center;}body{ background-color: #E6E6FA;}</style></head><body><br><br><br><h1>YOUR LOGIN IS SUCCESSFUL FOR BANK</h1></html>");**

**c=1;**

**}**

**}**

**if(c==0)**

**{out.println("<html><head><style>h1{text-align: center;}body{ background-color: #E6E6FA;}</style></head><body><br><br><br><h1>YOUR ARE NOT REGISTERED FOR OUR BANK REGISTER FIRST THEN LOGIN TO IT</h1></html>");**

**}**

**}**

**catch(Exception e)**

**{**

**System.out.println("ERROR:"+ e.getMessage());**

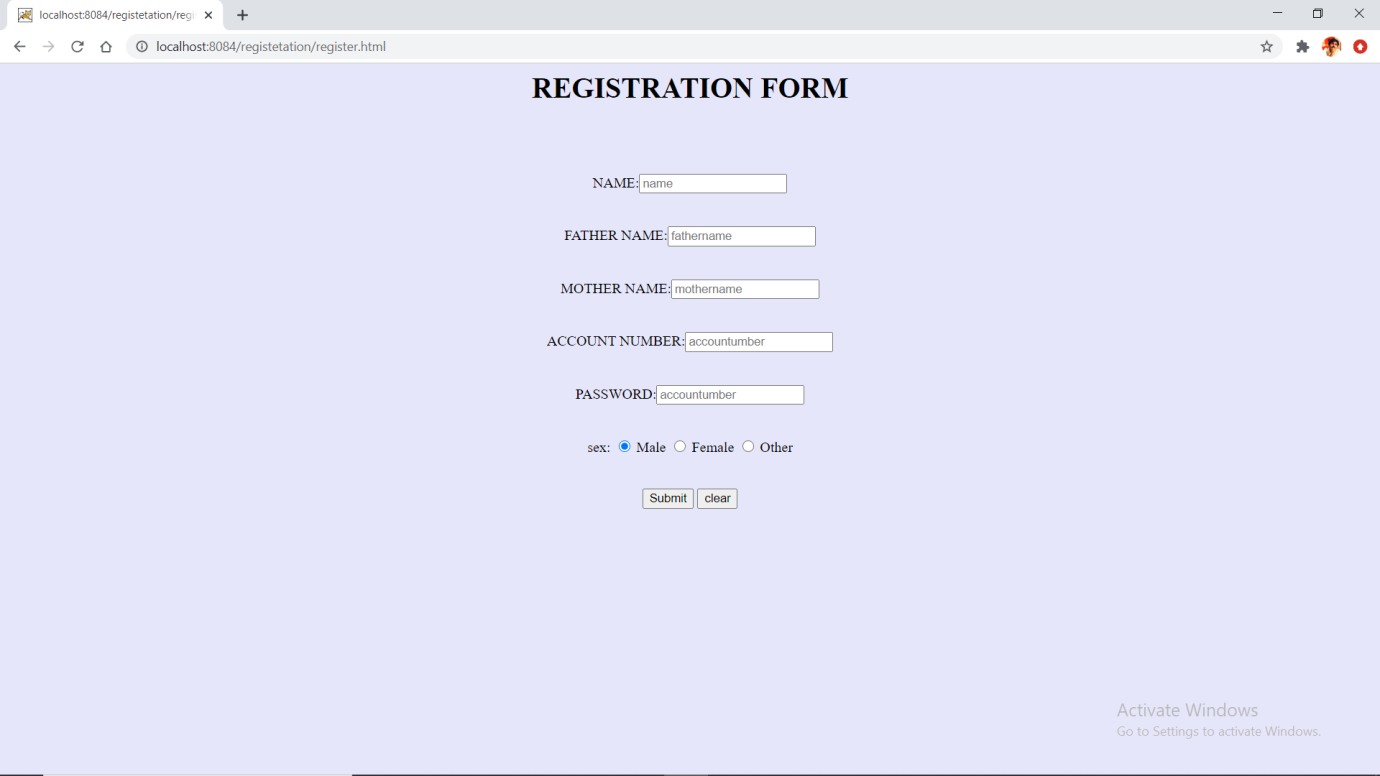
**}**

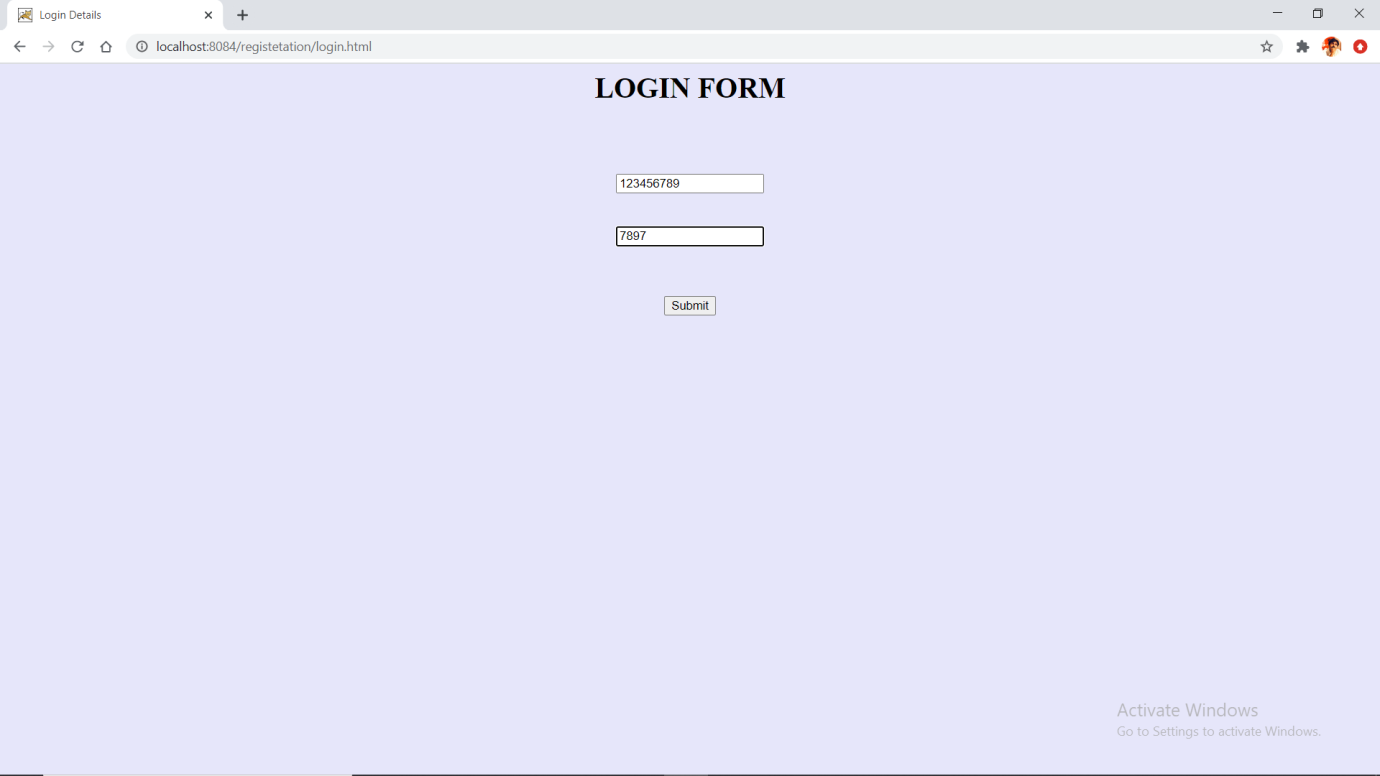
**}**

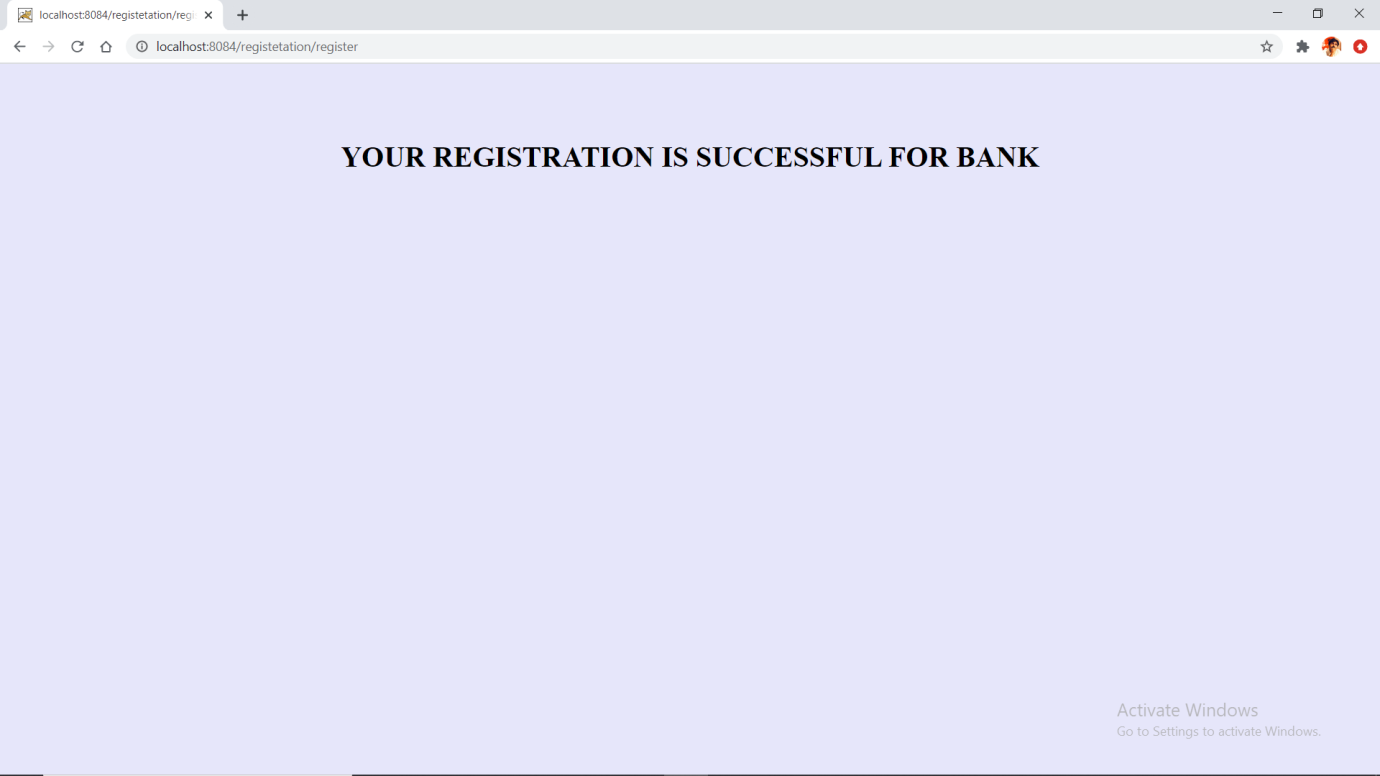
**}**

**6)OUTPUT:**

****

****

****

****

**7)CONCULSION:**

**Student able to check the results from these application using database .and the process of the application is very fast comparing to the application. These application is useful when ever the load is less.**

**8)REFERENCES:**

[**https://www.w3schools.com/**](https://www.w3schools.com/)

[**https://www.javatpoint.com/**](https://www.javatpoint.com/)