A picture containing logo

Description automatically generated

**Dynamic Web Projects using Java Spring Boot**

**Objective:**

1. To display customer data on the local web browser using Java Spring Boot, Tomcat Apache, Maven, MySQL database and Web Programming tools (HTML, CSS, JavaScript, JSP) – eg <http://localhost:8080/getPincode>
2. To host WAR file on a secure cloud Server (Amazon AWS, Veeble, Linode etc) and make the customer data available on a public URL by mapping the domain – eg <https://funwithprogramming.com/getPincode>

**Pre-Requisites:**

|  |  |  |
| --- | --- | --- |
| **SN** | **Tool** | **Remarks** |
| 1 | MySQL WorkBench, MySQL Server | For Database |
| 2 | Spring Tool Suite (STS) | For Java Programming |
| 3 | Notepad, Sublime Text or VS Code | For HTML and JSP web pages |

**Main Steps:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SN** | **Steps** | **Tool** | | **Remarks** | |
| 1 | Create Database “sangamone” | | MySQL Workbench | | Note1 |
| 2 | Create Table “pincode” | | MySQL Workbench | | Note2 |
| 3 | Import pincode.csv to pincode table | | MySQL Workbench | | Note3 |
| 4 | Verify DB – “select \* from pincode” | | MySQL Workbench | | Note4 |
| 5 | Create Spring Starter Project pincode2 | | STS | | Note5 |
| 6 | Enable Required Dependency | | STS | | Note6 |
| 7 | Create Pincode2Controller class in com.sangamone.controller package | | STS | | Note7 |
| 8 | Create Pincode2 class in com.sangamone.entity package | | STS | | Note8 |
| 9 | Create PincodeRepo interface in com.sangamone.repository package | | STS | | Note9 |
| 10 | Add in application.properties database driver name, database name, username, password | | STS | | Note10 |
| 11 | EntityClass Pincode2.java –   1. Add @Entity annotation outside the entity class 2. Add @Table annotation with name=”pincode” outside the entity class 3. Add @Id annotation inside Pincode entity class 4. Add @GeneratedValue(strategy=GenerationType.AUTO) inside Pincode entity class 5. Generate Constructor using Fields 6. Generate Constructor from Superclass 7. Generate Getters and Setters | | STS | | Note11 |
| 12 | Repository Interface Pincode2Repo.java   1. public interface Pincode2Repo extends JpaRepository<Pincode2, Integer> { } 2. Create method –   List<Pincode> getPincode(String pincode)   1. Add @Query (value=”select \* from pincode where pincode=?1”, nativeQuery=true) annotation just above the method | | STS | | Note12 |
| 13 | JSP Page creation under src – main – webapps   1. enterPincode.jsp – to enter a specific pincode 2. display.jsp – to display specific pincode entered in above jsp 3. displayAll.jsp – to display all pincodes without the need for any data input.   URLs are  <http://localhost:8080/enterPincode.jsp>  <http://localhost:8080/display.jsp>  <http://localhost:8080/displayAll.jsp> | | STS | | Note13 |
| 14 | JSP enterPincode.jsp  <form action=*"getPincode"* method=*"GET"*>  <input type=*"text"* name=*"pincode"* placeholder=*"enter pincode"*>  <button type=*"submit"*>Submit</button> | | STS | | Note14 |
| 15 | JSP display.jsp  <c:forEach var=*"pincode"* items = *"*${pincode}*"*>  <tr>  <td>${pincode.id}</td>  <td>${pincode.city}</td>  <td>${pincode.division}</td>  <td>${pincode.office}</td>  <td>${pincode.pincode}</td>  <td>${pincode.district}</td>  <td>${pincode.state}</td>  </tr>  </c:forEach> | | STS | | Note15 |
| 16 | JSP displayAll.jsp  <c:forEach var=*"pincode"* items = *"*${pincode}*"*>  <tr>  <td>${pincode.id}</td>  <td>${pincode.city}</td>  <td>${pincode.division}</td>  <td>${pincode.office}</td>  <td>${pincode.pincode}</td>  <td>${pincode.district}</td>  <td>${pincode.state}</td>  </tr>  </c:forEach> | | STS | | Note16 |
| 17 | Controller class pincode2controller.java   1. Add @Controller annotation outside the Controller class 2. Add @Autowired annotation inside the Controller class 3. Create object for Repository class – Pincode2Repo pincode2Repo 4. Add @RequestMapping annotation - @RequestMapping(“/”) 5. Create Method getPincode(Model model, @RequestParam int pincode){ model.addAttribute(“pincode”, pincode2Repo.getPincode(pincode)); return “display.jsp”;} 6. Add @RequestMapping annotation - @RequestMapping(“/”) 7. Create Method getPincodeAll(Model model){ model.addAttribute(“pincode”, pincode2Repo.findAll()); return “displayAll.jsp”;} | |  | | Note17 |
| 18 | Pom.xml -verify if dependencies are included  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-tomcat</artifactId>  <scope>provided</scope>  </dependency>  <dependency>  <groupId>javax.servlet</groupId>  <artifactId>jstl</artifactId>  </dependency>  <dependency>  <groupId>org.apache.tomcat</groupId>  <artifactId>tomcat-jasper</artifactId>  <version>9.0.54</version>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-data-jpa</artifactId>  <version>2.5.4</version>  </dependency> | |  | | Note18 |
| 19 | Testing:  <http://localhost:8080/displayAll.jsp>  Should show a list of all pincodes with details  <http://localhost:8080/enterPincode.jsp>  Should show a form with input for entering a specific pincode – eg. 577228  <http://localhost:8080/display.jsp>  Should show only specific pincodes which match the input pincode entered. | |  | | Note19 |
| 20 | Generate WAR file  Project – right-click – Export – WAR file  Before generating WAR file, change Database credential to that of Database in Veeble Cloud Server db.driverClassName=com.mysql.jdbc.Driver db.url=jdbc:mysql://localhost:3306/SpotFoto db.username=root db.password=BijleePoint hibernate.dialect=org.hibernate.dialect.MySQLDialect | |  | | Note20 |
| 21 | Go to Veeble (Cloud Hosting Provider)  <https://sangamone.com/cpanel>   * Sangamone3 * UniversalMobileBankingApp<UScurrency><Mooru> | |  | | Note21 |
| 22 | Veeble – FileManager – ea-tomcat85 – webapps  Upload WAR file here | |  | | Note22 |
| 23 | Veeble – phpMyAdmin – for DB changes – SQL - create database sangamone – use sangamone – create table pincode – similar to the one created in local. Alternately, the entire DB can be exported into pincode.sql. This SQL file can be run on Veeble Database Server | |  | | Note23 |
| 24 | Testing - <http://184.164.137.185:10000/Pincode2/enterPincode.jsp>  <http://sangamone.com:10000/Pincode2/enterPincode.jsp> | |  | | Note24 |
| 25 | Alternate cPanel: with Java8  <https://cs.sangamone.com:2087/cpsess9432451841/scripts/command?PFILE=cPanel>  username: root Useless: OmamPe51HA | |  | | Note25 |