CR Management System – Java Spring Boot Guide

1. Project Setup

1.1 Create a Spring Boot Project

- Use Spring Initializr (https://start.spring.io/) or manually create a Maven project.
- Dependencies needed:
- Spring Web (`spring-boot-starter-web`)
- Spring Data JPA (`spring-boot-starter-data-jpa`)
- MySQL Driver (`mysql-connector-j`)
- Spring Boot Test (`spring-boot-starter-test`)

1.2 Common Issues

- Cannot resolve symbol for dependencies: Missing Maven dependencies or no internet.
- Solution: Download JAR manually into `libs` or use internet connection.

2. Folder Structure

```
src
 └── main
     ├── java
     │    └── com.example.CRManagementSystem
     │          ├── controller
     │          │      └── LoginController.java
     │          ├── model
     │          │      └── User.java
     │          └── repository
     │                 └── UserRepository.java
     └── resources
          └── application.properties
```

2.1 Common Issues

- Package mismatch: Java package names must match folder paths exactly.

3. Entity Class (User.java)

```java
package com.example.CRManagementSystem.model;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;

@Entity
public class User {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String username;
    private String password;
    private String role;

    public User() {}
    public User(String username, String password, String role) {
        this.username = username;
        this.password = password;
        this.role = role;
    }

    // Getters and Setters
}
```

3.3 Common Issues

- Cannot resolve symbol `Entity` or `Id`: Missing `spring-boot-starter-data-jpa` dependency.

4. Repository Interface (UserRepository.java)

```
package com.example.CRManagementSystem.repository;

import com.example.CRManagementSystem.model.User;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface UserRepository extends JpaRepository<User, Long> {
    User findByUsernameAndPassword(String username, String password);
}
```

4.3 Common Issues

- Cannot resolve symbol 'repository': Repository package or interface doesn't exist or package declaration is incorrect.

5. Controller (LoginController.java)

```
package com.example.CRManagementSystem.controller;

import com.example.CRManagementSystem.model.User;
import com.example.CRManagementSystem.repository.UserRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

@RestController
@RequestMapping("/api")
@CrossOrigin(origins = "*")
public class LoginController {

    @Autowired
    private UserRepository userRepository;

    @PostMapping("/login")
    public String loginUser(@RequestBody User userData) {
        User user =
userRepository.findByUsernameAndPassword(userData.getUsername(),
userData.getPassword());

        if (user == null) {
            return "Invalid username or password!";
        } else {
            return "Login successful!";
        }
    }
}
```

5.3 Common Issues

- Cannot autowire `UserRepository`: Repository class/package not created or package mismatch.

6. application.properties

```
spring.application.name=CRManagementSystem
server.port=8080

spring.datasource.url=jdbc:mysql://172.27.128.60:3306/cr_db?
useSSL=false&allowPublicKeyRetrieval=true&serverTimezone=UTC
spring.datasource.username=root
spring.datasource.password=admin123
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
```

6.3 Common Issues

- Cannot connect to MySQL: IP, port, username, password wrong or remote access not allowed.
- Solution: Grant remote access in MySQL:

  ```
  GRANT ALL PRIVILEGES ON cr_db.* TO 'root'@'%' IDENTIFIED BY 'admin123';
  FLUSH PRIVILEGES;
  ```

7. Frontend HTML Integration

- Place login HTML in `src/main/resources/static`.
- Change form action to call Spring Boot API using AJAX.

```
<form id="loginForm">
    <input type="text" name="username">
    <input type="password" name="password">
    <button type="submit">Login</button>
</form>
<script>
document.getElementById('loginForm').addEventListener('submit', function(e) {
    e.preventDefault();
    fetch('http://localhost:8080/api/login', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify({
            username: this.username.value,
            password: this.password.value
        })
    })
    .then(res => res.text())
    .then(alert);
});
</script>
```

8. Maven pom.xml Key Points

- Include dependencies:

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<dependency>
    <groupId>com.mysql</groupId>
    <artifactId>mysql-connector-j</artifactId>
    <version>8.1.0</version>
    <scope>system</scope>
    <systemPath>${project.basedir}/libs/mysql-connector-j-8.1.0.jar</
systemPath>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

9. Common Errors and Fixes

Error	Cause	Solution
Cannot resolve symbol `repository`	Repository class/package missing or package name mismatch	Create repository and match package names
Cannot resolve symbol `Entity`	Missing JPA dependency	Add `spring-boot-starter-data-jpa` dependency
Dependency not found	Maven cannot download JAR	Use manual JAR in `libs` folder
Database connection fails	Wrong URL/IP/user/password or remote access not allowed	Verify MySQL server and grant remote privileges