

Java EndSem Practical Exam

Name-NAITIK KUKREJA RollNO-202401100400120

<https://github.com/Naitik4141/java-praCTICAL>

UserController.java

1. Model

package com.example.usermanagement.entity;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

import lombok.Data;

import lombok.NoArgsConstructor;

import lombok.AllArgsConstructor;

@Entity

@Table(name = "users")

@Data

@NoArgsConstructor

@AllArgsConstructor

public class User {

 @Id

 @GeneratedValue(strategy = GenerationType.IDENTITY)

```
private Long id;  
  
private String username;  
private String email;  
}
```

2.View(FrontEnd)

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="UTF-8">  
<meta name="viewport" content="width=device-width, initial-  
scale=1.0">  
<title>User Management System</title>  
<link rel="stylesheet" href="style.css">  
<link  
href="https://fonts.googleapis.com/css2?family=Inter:wght@400;  
500;600;700&display=swap" rel="stylesheet">  
</head>  
<body>  
<div class="container">  
<header>  
<h1>User Management</h1>  
<p>Manage your users efficiently and intuitively.</p>  
</header>  
  
<main>  
<section class="card form-section">
```

```
<h2>Add New User</h2>
<form id="userForm">
  <input type="hidden" id="userId">
  <div class="form-group">
    <label for="username">Username</label>
    <input type="text" id="username" placeholder="e.g.
johnndoe" required>
  </div>
  <div class="form-group">
    <label for="email">Email Address</label>
    <input type="email" id="email" placeholder="e.g.
john@example.com" required>
  </div>
  <div class="form-actions">
    <button type="submit" id="submitBtn" class="btn
primary">
      <span class="icon">+</span> Add User
    </button>
    <button type="button" id="cancelBtn" class="btn
secondary hidden">Cancel Edit</button>
  </div>
</form>
</section>
```

```
<section class="card list-section">
  <div class="list-header">
    <h2>Current Users</h2>
    <span class="badge" id="userCount">0 Users</span>
```

```
____</div>
____<div class="table-container">
____<table id="userTable">
____<thead>
____<tr>
____<th>ID</th>
____<th>Username</th>
____<th>Email</th>
____<th>Actions</th>
____</tr>
____</thead>
____<tbody>
____<!-- User rows will be injected here -->
____</tbody>
____</table>
____<div id="emptyState" class="empty-state">
____<p>No users found. Start by adding one above.</p>
____</div>
____</div>
____</div>
____</section>
____</main>
____</div>
```

```
____<div id="toast" class="toast hidden">Action Successful</div>
____<script src="app.js"></script>
____</body>
____</html>
```

3.Controller(API)

package com.example.usermanagement.controller;

import com.example.usermanagement.entity.User;

import com.example.usermanagement.service.UserService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController

@RequestMapping("/users")

public class UserController {

 private final UserService userService;

 @Autowired

 public UserController(UserService userService) {

 this.userService = userService;

 }

 @PostMapping

 public User createUser(@RequestBody User user) {

 return userService.createUser(user);

 }

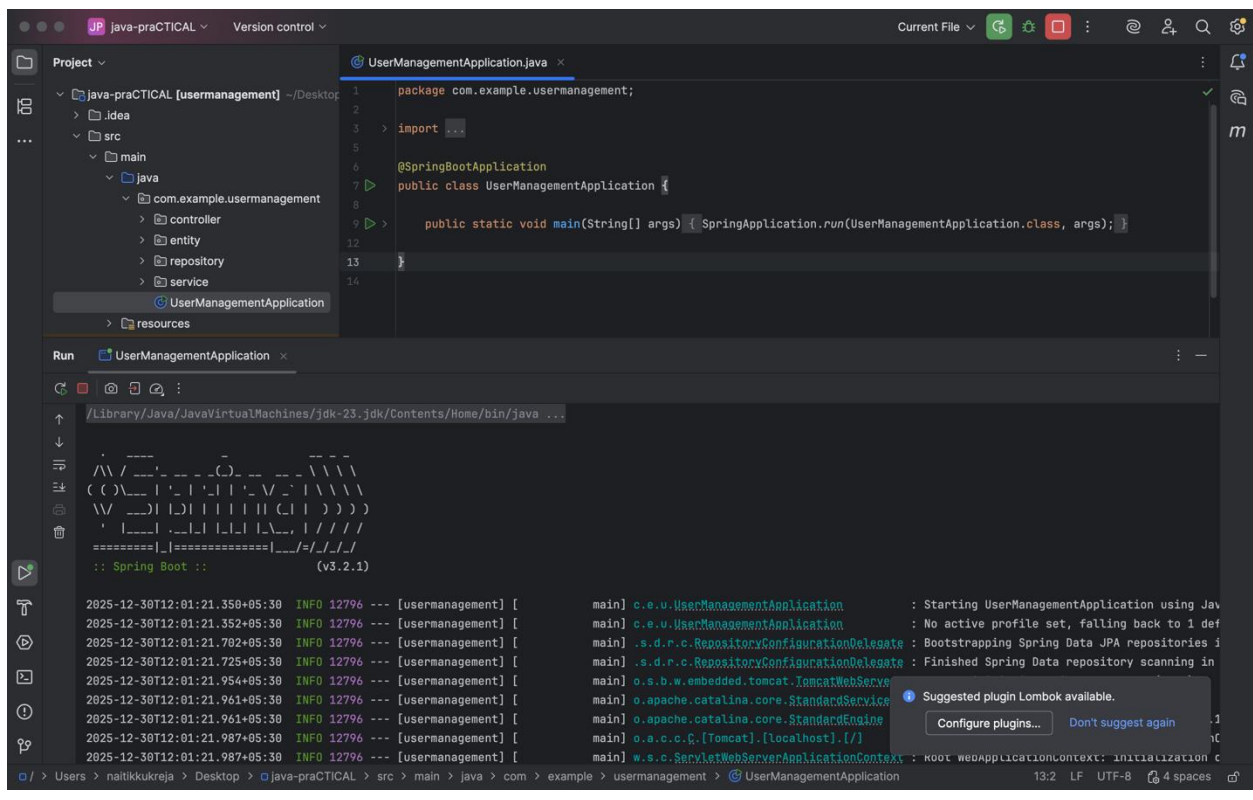
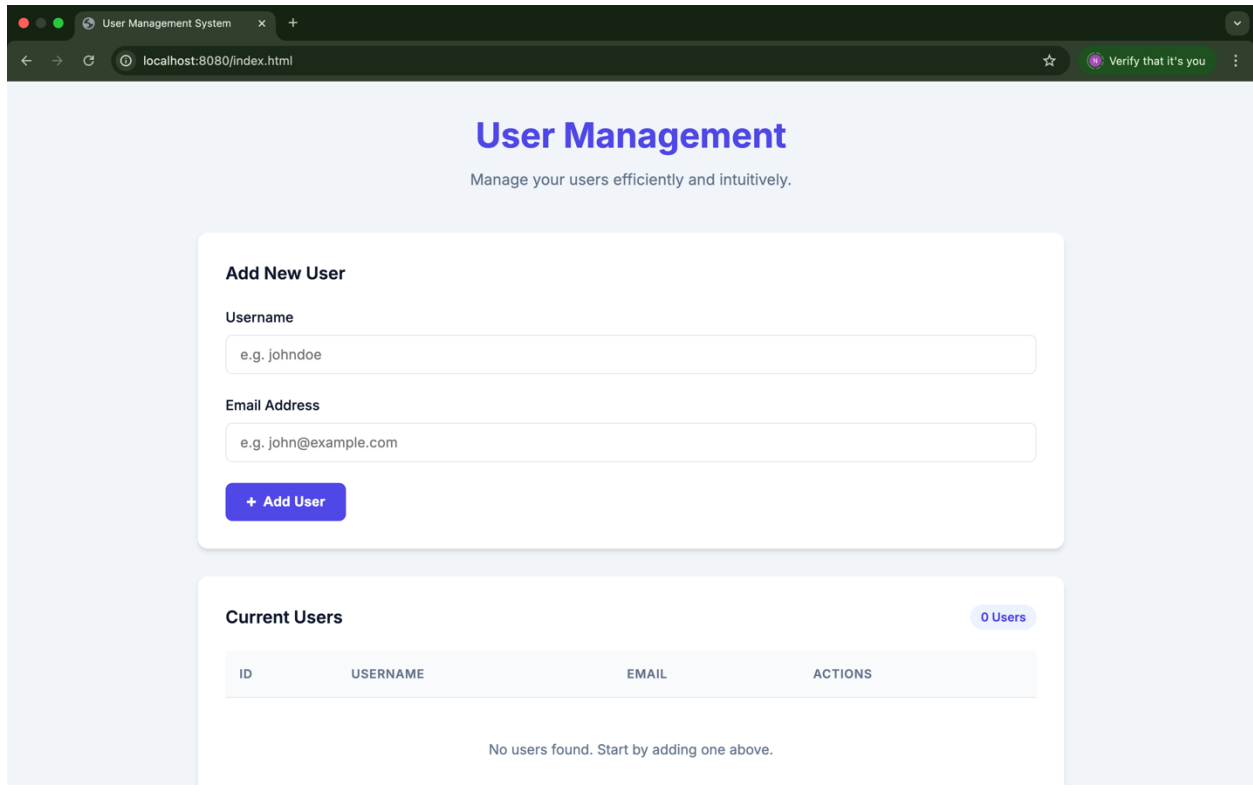
 @GetMapping

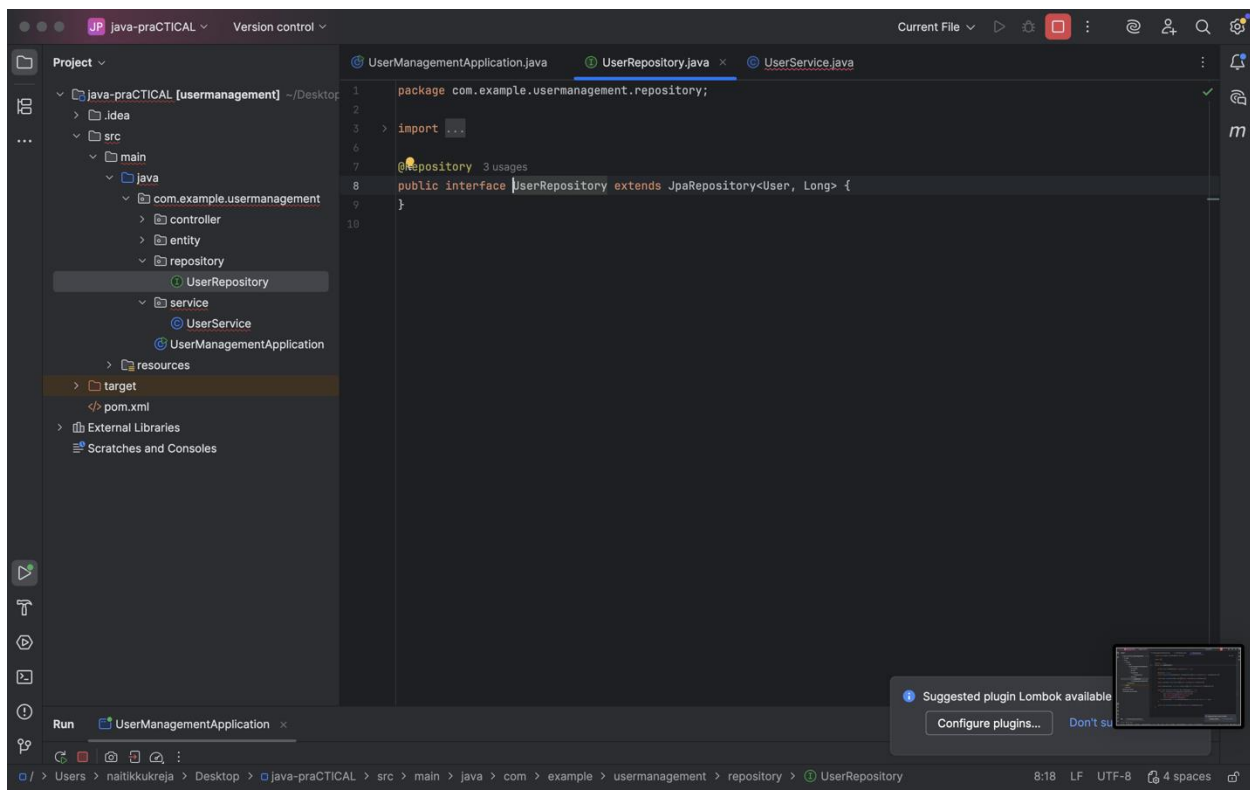
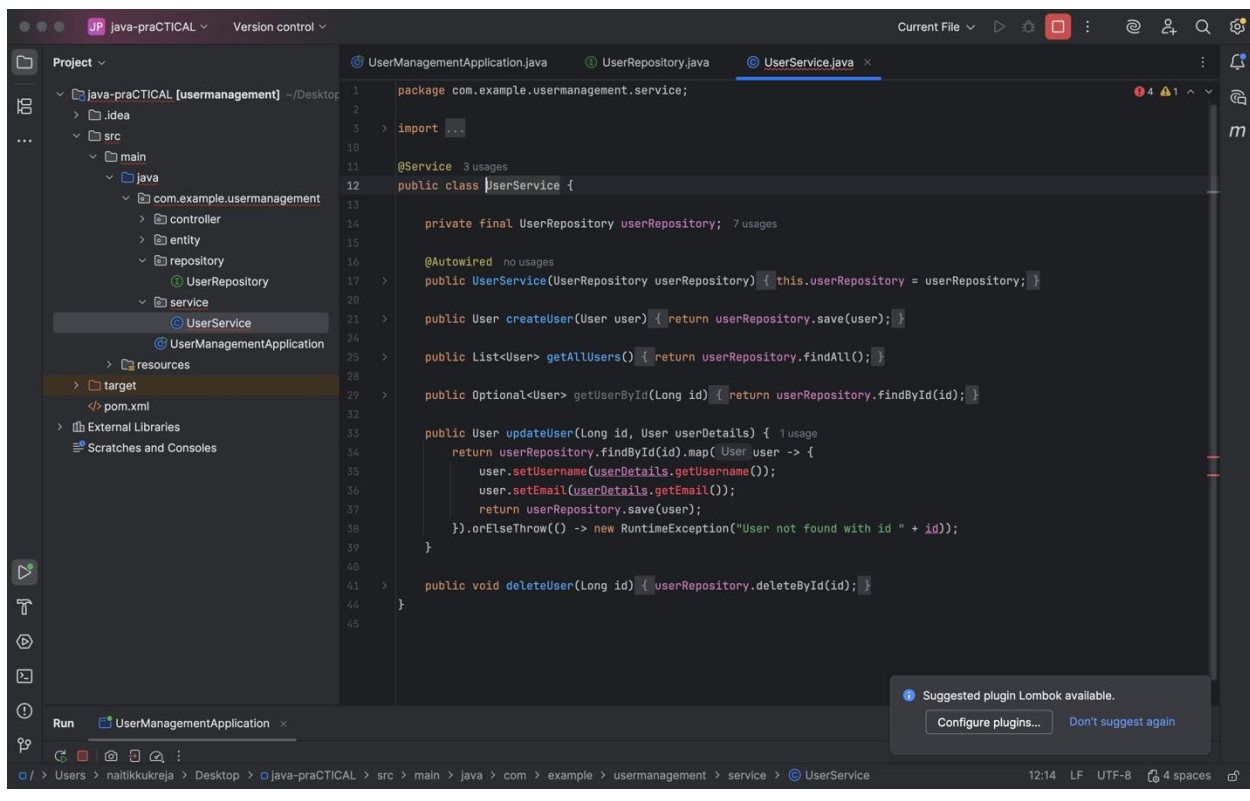
 public List<User> getAllUsers() {

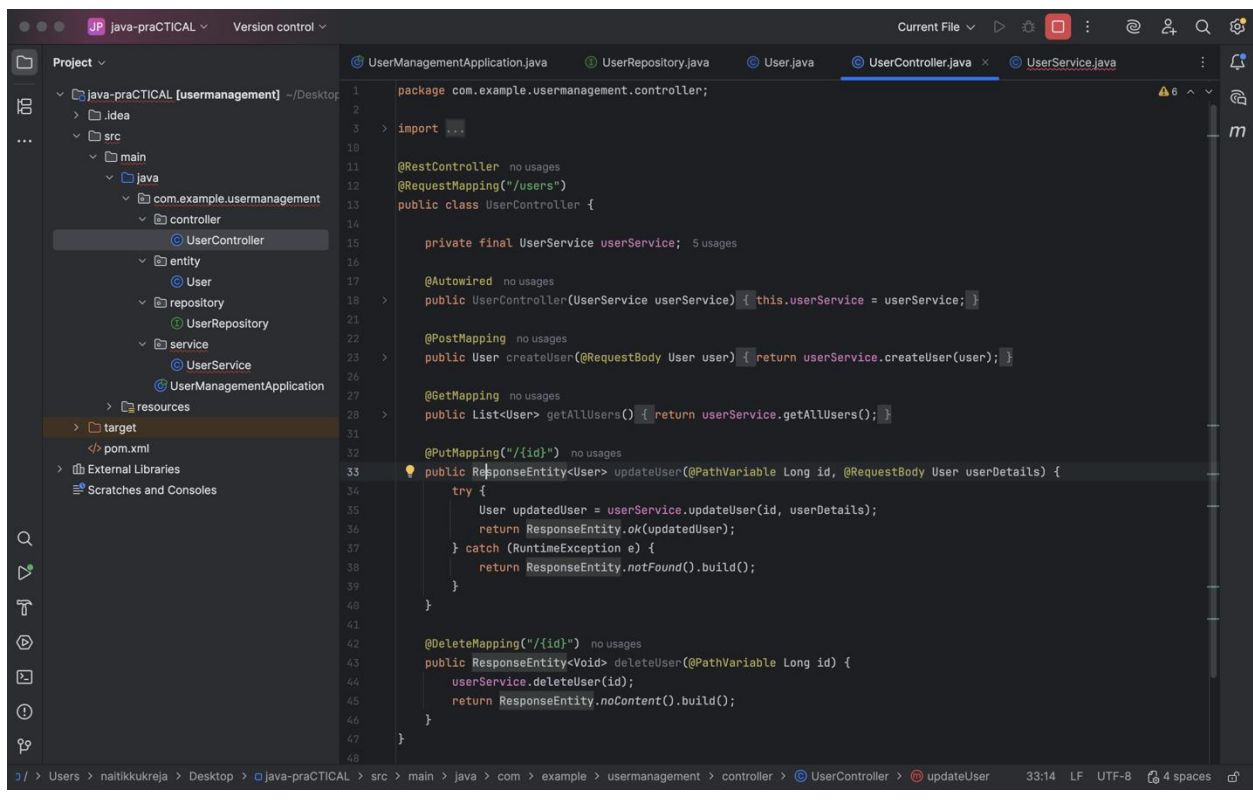
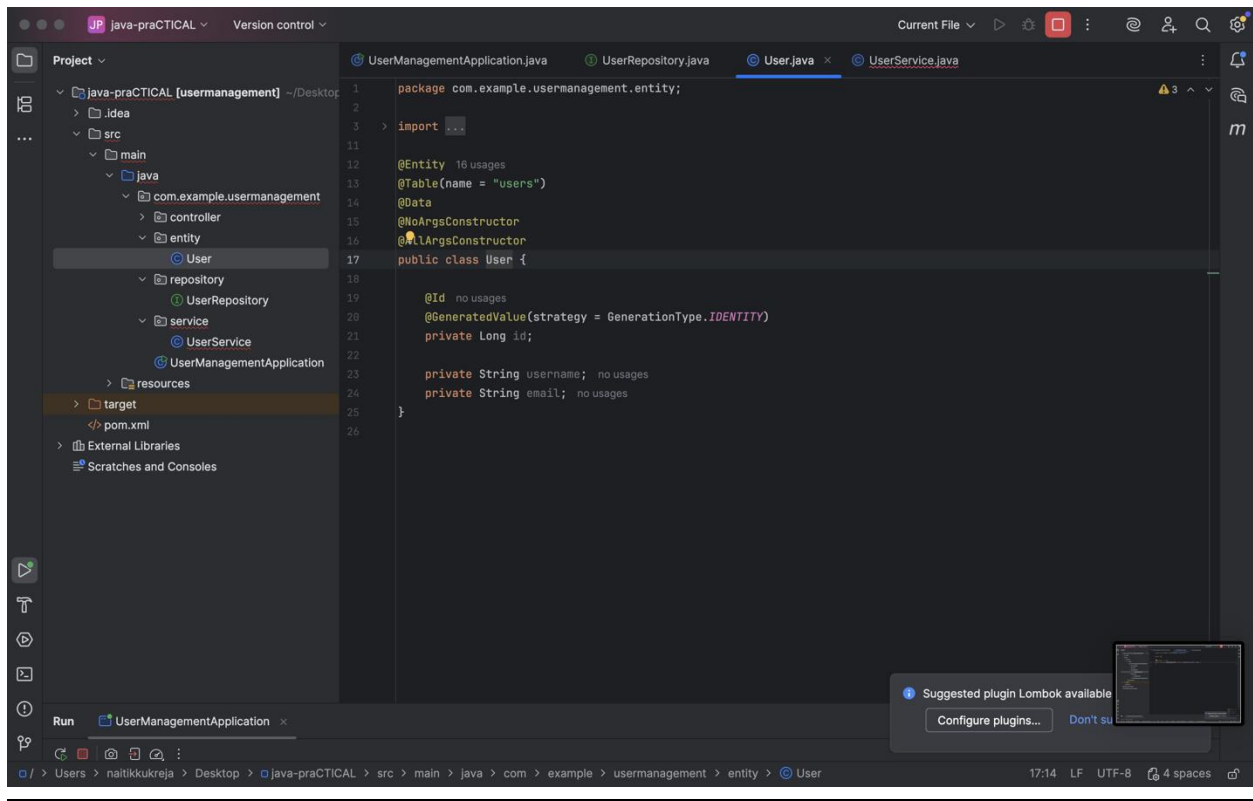
```
    return userService.getAllUsers();  
}
```

```
@PutMapping("/{id}")  
public ResponseEntity<User> updateUser(@PathVariable Long  
id, @RequestBody User userDetails) {  
    try {  
        User updatedUser = userService.updateUser(id,  
userDetails);  
        return ResponseEntity.ok(updatedUser);  
    } catch (RuntimeException e) {  
        return ResponseEntity.notFound().build();  
    }  
}
```

```
@DeleteMapping("/{id}")  
public ResponseEntity<Void> deleteUser(@PathVariable Long  
id) {  
    userService.deleteUser(id);  
    return ResponseEntity.noContent().build();  
}  
}
```







The screenshot shows the IntelliJ IDEA interface with the 'index.html' file open in the editor. The project structure on the left includes 'java-praCTICAL [usermanagement]' with subfolders like '.idea', 'src', 'main', 'resources', and 'static'. The 'index.html' file is located in the 'static' folder. The code in the editor is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>User Management System</title>
7   <link rel="stylesheet" href="style.css">
8   <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700&display=swap" rel="stylesheet">
9 </head>
10 <body>
11   <div class="container">
12     <header>
13       <h1>User Management</h1>
14       <p>Manage your users efficiently and intuitively.</p>
15     </header>
16
17     <main>
18       <section class="card form-section">
19         <h2>Add New User</h2>
20         <form id="userForm">
21           <input type="hidden" id="userId">
22           <div class="form-group">
23             <label for="username">Username</label>
24             <input type="text" id="username" placeholder="e.g. johndoe" required>
25           </div>
26           <div class="form-group">
27             <label for="email">Email Address</label>
28             <input type="email" id="email" placeholder="e.g. john@example.com" required>
29           </div>
30           <div class="form-actions">
31             <button type="submit" id="submitBtn" class="btn primary">
32               <span class="icon"></span> Add User
33             </button>
34             <button type="button" id="cancelBtn" class="btn secondary hidden">Cancel Edit</button>
35           </div>
36         </form>
37       </section>
38     </main>
39   </div>
40 </body>
41 </html>
```

The screenshot shows the IntelliJ IDEA interface with the 'application.properties' file open in the editor. A notification banner at the top states 'Spring Boot configuration files are supported by IntelliJ IDEA Ultimate'. The code in the editor is as follows:

```
1 spring.application.name=usermanagement
2 spring.datasource.url=jdbc:h2:mem:testdb
3 spring.datasource.driverClassName=org.h2.Driver
4 spring.datasource.username=sa
5 spring.datasource.password=password
6 spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
7 spring.h2.console.enabled=true
8 server.port = 8081
9
```