**JWT Assignment Problems**

### Exercise: Create an Authentication Service that Returns JWT

**Objective:** Create a Spring Boot authentication service that returns a JWT upon successful login.

### Step 1: Add Dependencies in pom.xml

<dependency>  
 <groupId>io.jsonwebtoken</groupId>  
 <artifactId>jjwt</artifactId>  
 <version>0.9.1</version>  
</dependency>  
<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-security</artifactId>  
</dependency>

### Step 2: Create User Model

public class User {  
 private String username;  
 private String password;  
 // getters and setters  
}

### Step 3: JWT Utility Class

@Component  
public class JwtUtil {  
 private String secret = "secret\_key";  
  
 public String generateToken(String username) {  
 return Jwts.builder()  
 .setSubject(username)  
 .setIssuedAt(new Date())  
 .setExpiration(new Date(System.currentTimeMillis() + 1000 \* 60 \* 60 \* 10))  
 .signWith(SignatureAlgorithm.HS256, secret).compact();  
 }  
}

### Step 4: Authentication Controller

@RestController  
public class AuthController {  
  
 @Autowired  
 private JwtUtil jwtUtil;  
  
 @PostMapping("/authenticate")  
 public ResponseEntity<String> authenticate(@RequestBody User user) {  
 if ("admin".equals(user.getUsername()) && "password".equals(user.getPassword())) {  
 String token = jwtUtil.generateToken(user.getUsername());  
 return ResponseEntity.ok(token);  
 } else {  
 return ResponseEntity.status(HttpStatus.UNAUTHORIZED).body("Invalid credentials");  
 }  
 }  
}

### Step 5: Application Properties

server.port=8080

### Sample Input (POST /authenticate):

{  
 "username": "admin",  
 "password": "password"  
}

### Sample Output:

eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJhZG1pbiIsImlhdCI6MTY5MDQxMDY2MywiZXhwIjoxNjkwNDE0MjYzfQ.LTt3wL7zA5FScKqK\_TJZgguMmvBzUzA4rx7mXSUmsOE