

Inventra - Intelligent Inventory Management System

Presented by
RAJESWARI R

ABOUT

- **Definition:** Inventra is an intelligent repository management system.
- **Purpose:** Helps businesses efficiently manage stock, suppliers, purchase orders, and sales analysis.
- **Key benefit:** Reduces manual effort and improves inventory accuracy.

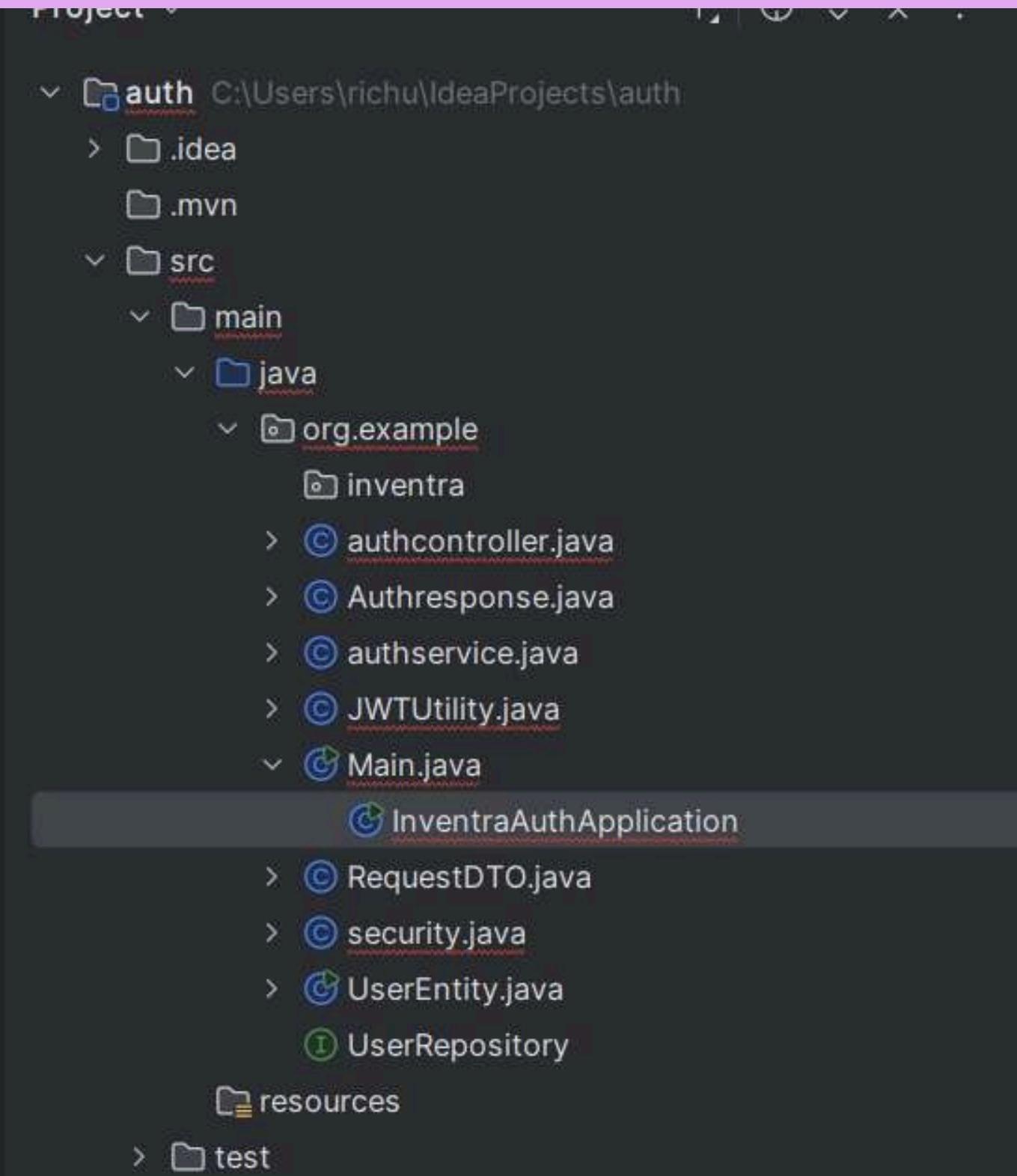
DB schema

Field Name	Data Type	Description
id	INT (Primary Key, Auto Increment)	Unique user ID
username	VARCHAR(50)	User login name
email	VARCHAR(100)	User email address
password	VARCHAR(255)	Hashed password
role	VARCHAR(20)	User role (Admin, Staff, Viewer)
created_at	TIMESTAMP	Account creation time
updated_at	TIMESTAMP	Last update time

Auth DTOs

DTO Class	Purpose	Fields	Used for
LoginRequest	Captures the credentials sent by a client trying to log in.	username, password	The POST /api/auth/login endpoint.
RegisterRequest	Captures all necessary data for creating a new user account.	username, email, password, role (optional)	The POST /api/auth/register endpoint.
ResetPasswordRequest	Captures the information needed to securely update a forgotten password.	email, newPassword	The POST /api/auth/reset-password endpoint.
AuthResponse	The standardized structure for the server's response after a successful login or registration.	username, role, token (the generated JWT)	Returned by the login and register endpoints.

main classes



```
>Main.java  UserEntity.java  UserRepository.java  Authresponse.java  authservice.java  JWTUtility.java
1 package com.inventra.entity;
2
3 import jakarta.persistence.*;
4 import lombok.*;
5 import org.hibernate.annotations.CreationTimestamp;
6
7 import java.time.Instant;
8
9 @Entity 9 usages 2 related problems
10 @Table(name = "users")
11 @Data
12 @NoArgsConstructor
13 @AllArgsConstructor
14 @Builder
15 public class User {
16
17     @Id no usages
18     @GeneratedValue(strategy = GenerationType.IDENTITY)
19     @Column(name = "user_id")
20     private Long userId;
21
22     @Column(unique = true, nullable = false, length = 50) no usages
23     private String username;
24
25     @Column(unique = true, nullable = false, length = 100) no usages
26     private String email;
27
28     @Column(nullable = false) no usages
29     private String password;
30
31     @Column(length = 30) no usages
32     private String role; // ADMIN, STAFF, MANAGER
```

```
1 package com.inventra.repository;
2
3 import com.inventra.entity.User;
4 import org.springframework.data.jpa.repository.JpaRepository;
5 import org.springframework.stereotype.Repository;
6
7 import java.util.Optional;
8
9 @Repository no usages 1 related problem
10 public interface UserRepository extends JpaRepository<User, Long> {
11     Optional<User> findByUsername(String username); no usages 1 related problem
12     Optional<User> findByEmail(String email); no usages 1 related problem
13     boolean existsByUsername(String username); no usages 1 related problem
14     ! boolean existsByEmail(String email); no us
15 }
```

```
>Main.java  UserEntity.java  UserRepository.java  Authresponse.java  authservice.java  JWTUtility.java
4 import com.inventra.entity.User;
5 import com.inventra.repository.UserRepository;
6 import com.inventra.util.JwtUtil;
7 import lombok.RequiredArgsConstructor;
8 import org.springframework.security.crypto.password.PasswordEncoder;
9 import org.springframework.stereotype.Service;
10
11 @Service no usages 1 related problem
12 @RequiredArgsConstructor
13 public class AuthService {
14
15     private final UserRepository userRepository; 6 usages
16     private final PasswordEncoder passwordEncoder; 3 usages
17     private final JwtUtil jwtUtil; 2 usages
18
19     @
20     public AuthResponse login(LoginRequest request) { no usages 2 related problems
21         User user = userRepository.findByUsername(request.getUsername())
22             .orElseThrow(() -> new RuntimeException("User not found"));
23
24         if (!user.getIsActive()) {
25             throw new RuntimeException("Account is inactive");
26         }
27
28         if (!passwordEncoder.matches(request.getPassword(), user.getPassword())) {
29             throw new RuntimeException("Invalid credentials");
30         }
31
32         String token = jwtUtil.generateToken(user.getUsername(), user.getRole());
33
34         return AuthResponse.builder()
35             .username(user.getUsername())
36             .role(user.getRole())
37     }
38 }
```

```
>Main.java  UserEntity.java  UserRepository.java  Authresponse.java  authservice.java  JWTUtility.java
1 package com.inventra;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 @SpringBootApplication
7 public class InventraAuthApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(InventraAuthApplication.class, args);
11     }
12
13 }
```

```
Main.java ① UserEntity.java ② UserRepository.java ③ Authresponse.java ④ authservice.java ⑤ JWTUtility.java × ▾  
1 package com.inventra.util;  
2  
3 import io.jsonwebtoken.JwtException;  
4 import io.jsonwebtoken.SignatureAlgorithm;  
5 import io.jsonwebtoken.io.Decoders;  
6 import io.jsonwebtoken.security.Keys;  
7 import org.springframework.stereotype.Component;  
8  
9 import java.security.Key;  
10 import java.util.Date;  
11 import java.util.HashMap;  
12 import java.util.Map;  
13  
14 @Component no usages 1 related problem  
15 public class JwtUtil {  
16  
17     // 256-bit HEX secret key for HS256 algorithm  
18     public static final String SECRET = "5367566859703373367639792F423F4528482B40625165546857605A71347437"; 1 usage  
19  
20     // 15 minutes in milliseconds  
21     private static final long EXPIRATION_TIME = 1000 * 60 * 15; 1 usage  
22  
23     public String generateToken(String username, String role) { no usages 2 related problems.  
24         Map<String, Object> claims = new HashMap<>();  
25         claims.put("role", role);  
26         return createToken(claims, username);  
27     }  
28  
29     private String createToken(Map<String, Object> claims, String subject) { 1 usage  
30         return Jwts.builder()  
31             .setClaims(claims)  
32             .setSubject(subject)
```

```
① Main.java ② UserEntity.java ③ UserRepository.java ④ Authresponse.java ⑤ authservice.java ⑥ JWTUtility.java ⑦ RequestDTO.java × ▾  
19  
20     // 15 minutes in milliseconds  
21     private static final long EXPIRATION_TIME = 1000 * 60 * 15; 1 usage  
22  
23     public String generateToken(String username, String role) { no usages 2 related problems.  
24         Map<String, Object> claims = new HashMap<>();  
25         claims.put("role", role);  
26         return createToken(claims, username);  
27     }  
28  
29     private String createToken(Map<String, Object> claims, String subject) { 1 usage  
30         return Jwts.builder()  
31             .setClaims(claims)  
32             .setSubject(subject)  
33             .setIssuedAt(new Date(System.currentTimeMillis()))  
34             .setExpiration(new Date(System.currentTimeMillis() + EXPIRATION_TIME))  
35             .signWith(getSignKey(), SignatureAlgorithm.HS256)  
36             .compact();  
37     }  
38  
39     private Key getSignKey() { 1 usage  
40         return Keys.hmacShaKeyFor("notThisKauf"); 1 usage  
41     }  
42  
43  
RequestDTO.java ① UserEntity.java ② UserRepository.java ③ Authresponse.java ④ authservice.java ⑤ JWTUtility.java ⑥ RequestDTO.java × ▾  
1 package com.inventra.dto;  
2  
3 import lombok.Data;  
4  
5 // Containing multiple DTOs in one file for concise copying;  
6 // in production, these can be split into separate files.  
7  
8 @Data no usages  
9 public class AuthRequests {  
10     // Placeholder if needed  
11 }  
12  
13 @Data no usages 2 related problems  
14 class LoginRequest {  
15     private String username; no usages  
16     private String password; no usages  
17 }  
18  
19 @Data no usages 2 related problems  
20 class RegisterRequest {  
21     private String username; no usages  
22     private String email; no usages  
23     private String password; no usages  
24     private String role; no usages  
25 }  
26  
27 @Data no usages 2 related problems  
28 class ResetPasswordRequest {  
29     private String email; no usages  
30     private String newPassword; no usages  
31 }
```

THANK YOU