Create authentication service that returns JWT

JwtUtil.java:

```
package com.cognizant.spring_learn.util;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import io.jsonwebtoken.security.Keys;

import org.springframework.stereotype.Component;

import java.security.Key;

import java.util.Date;

@Component

public class JwtUtil {

   private final Key key = Keys.hmacShaKeyFor("your-256-bit-secret-key-your-256-bit-secret-key".getBytes());

   private final long validityInMs = 3600000;

   public String generateToken(String username) {

      return Jwts.builder()

            .setSubject(username)

            .setIssuedAt(new Date())

            .setExpiration(new Date(System.currentTimeMillis() + validityInMs))

            .signWith(key, SignatureAlgorithm.HS256)

            .compact();

   }

}
```

AuthenticationController.java:

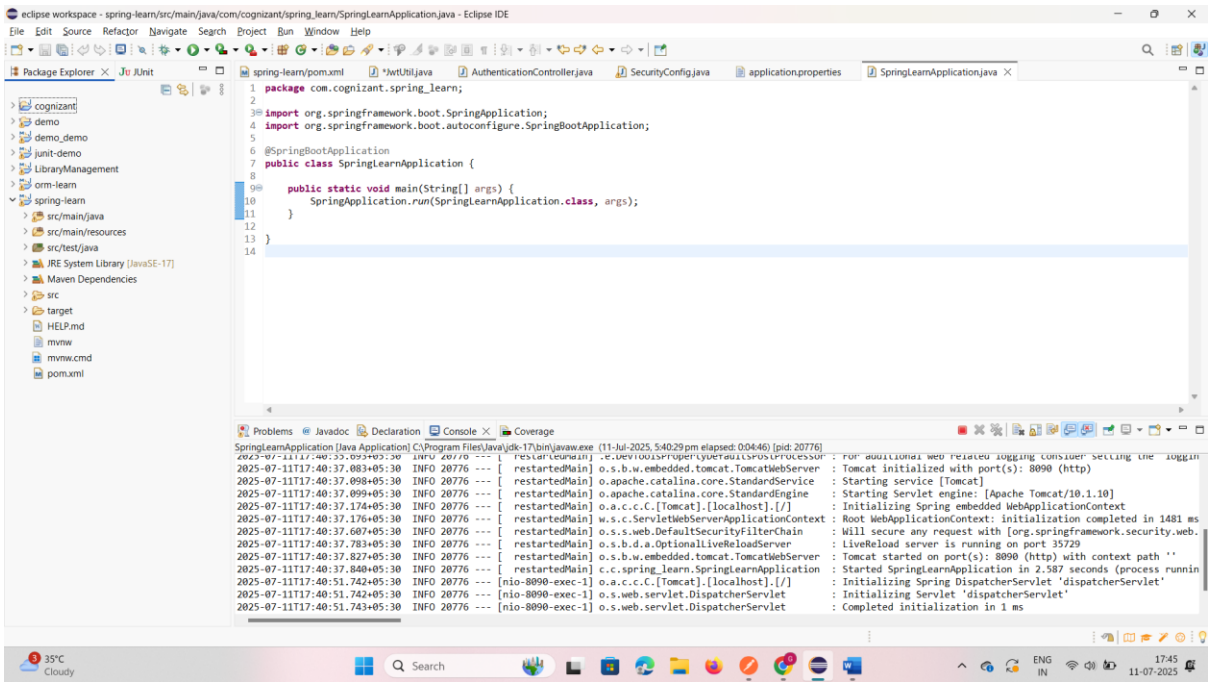
```
package com.cognizant.spring_learn.controller;
import com.cognizant.spring_learn.util.JwtUtil;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.Base64;
@RestController
public class AuthenticationController {
   @Autowired
   private JwtUtil jwtUtil;
   @RequestMapping(value = "/authenticate", method = RequestMethod.GET)
   public ResponseEntity<?> authenticate(@RequestHeader("Authorization") String authHeader) {
      if (authHeader != null && authHeader.startsWith("Basic ")) {
         String base64Credentials = authHeader.substring("Basic ".length());
         byte[] decodedBytes = Base64.getDecoder().decode(base64Credentials);
         String credentials = new String(decodedBytes);
         String[] values = credentials.split(":", 2);
         String username = values[0];
         String password = values[1];
         if ("user".equals(username) && "pwd".equals(password)) {
            String token = jwtUtil.generateToken(username);
            return ResponseEntity.ok().body("{\"token\":\"" + token + "\"}");
         } else {
            return ResponseEntity.status(401).body("Invalid Credentials");
         }
      } else {
         return ResponseEntity.status(400).body("Missing or invalid Authorization header") }
```

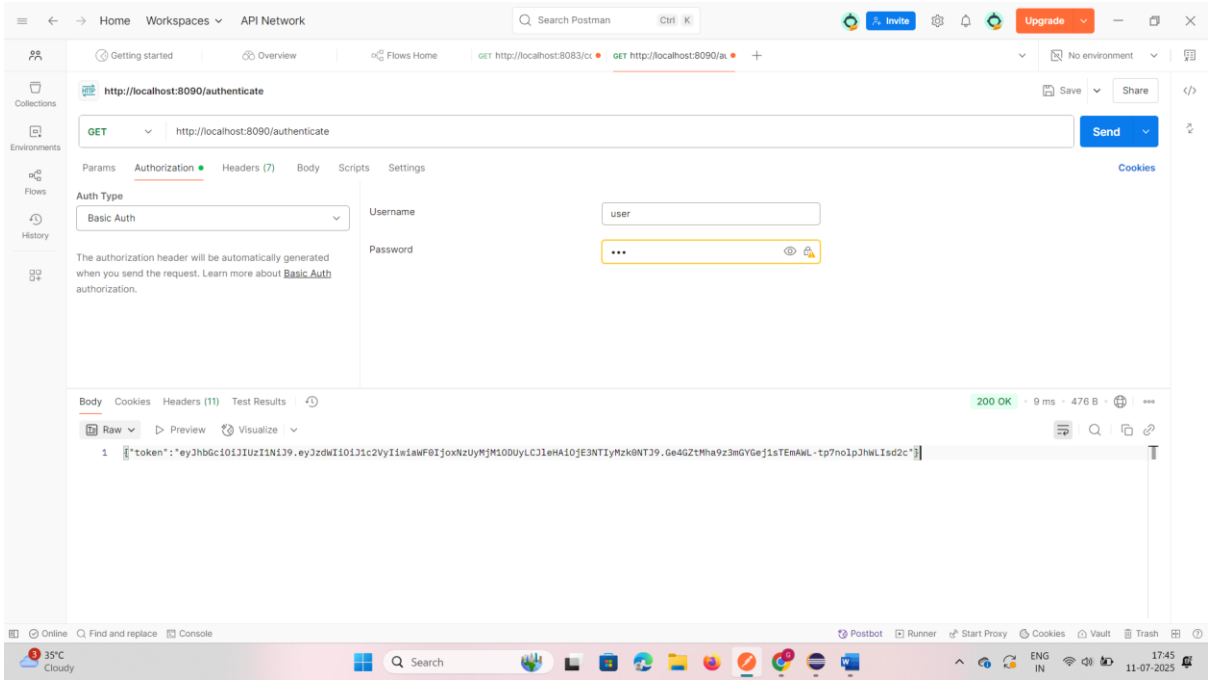
SecurityConfig.java:

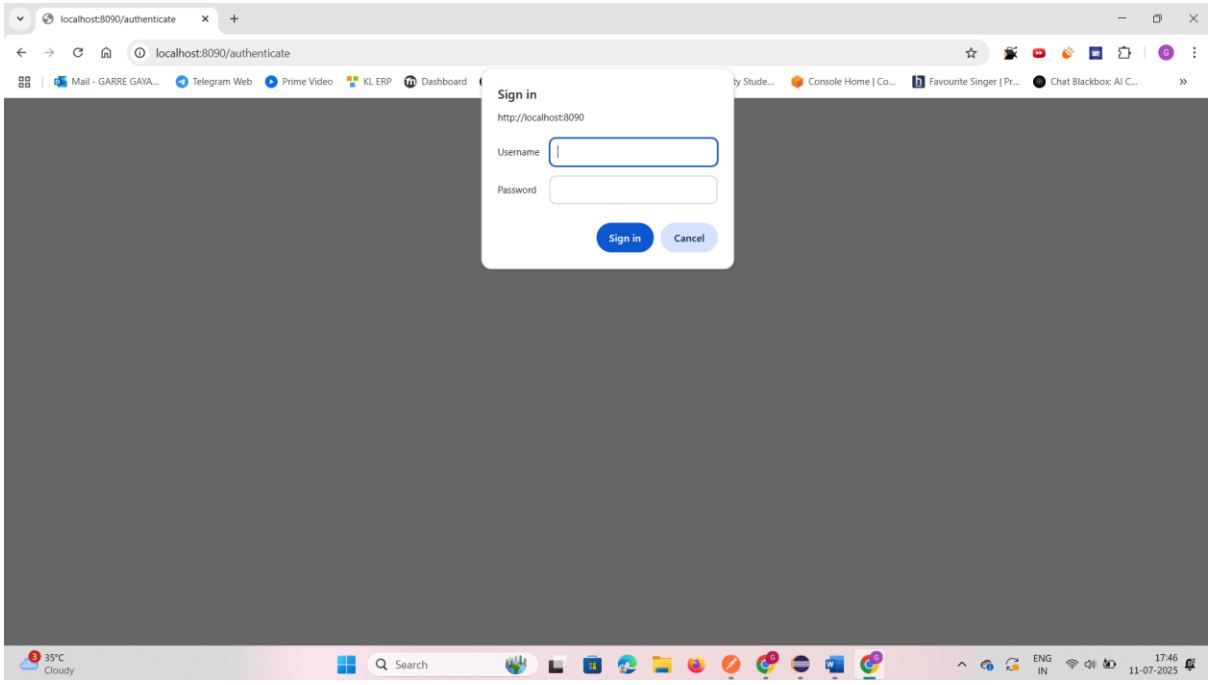
```
package com.cognizant.spring_learn.config;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.core.userdetails.User;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.provisioning.InMemoryUserDetailsManager;
import org.springframework.security.crypto.password.NoOpPasswordEncoder; // Only for testing!
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.security.web.SecurityFilterChain;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
@Configuration
public class SecurityConfig {
    @Bean
    public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
        http
            .csrf(csrf -> csrf.disable())
            .authorizeHttpRequests(auth -> auth
                .requestMatchers("/authenticate").permitAll()
                .anyRequest().authenticated()
            )
            .httpBasic(); // Enable Basic Auth
        return http.build();
    }
    @Bean
    public UserDetailsService userDetailsService() {
        return new InMemoryUserDetailsManager(
            User.withUsername("user")
                .password("pwd")
                .roles("USER")
                .build()
```

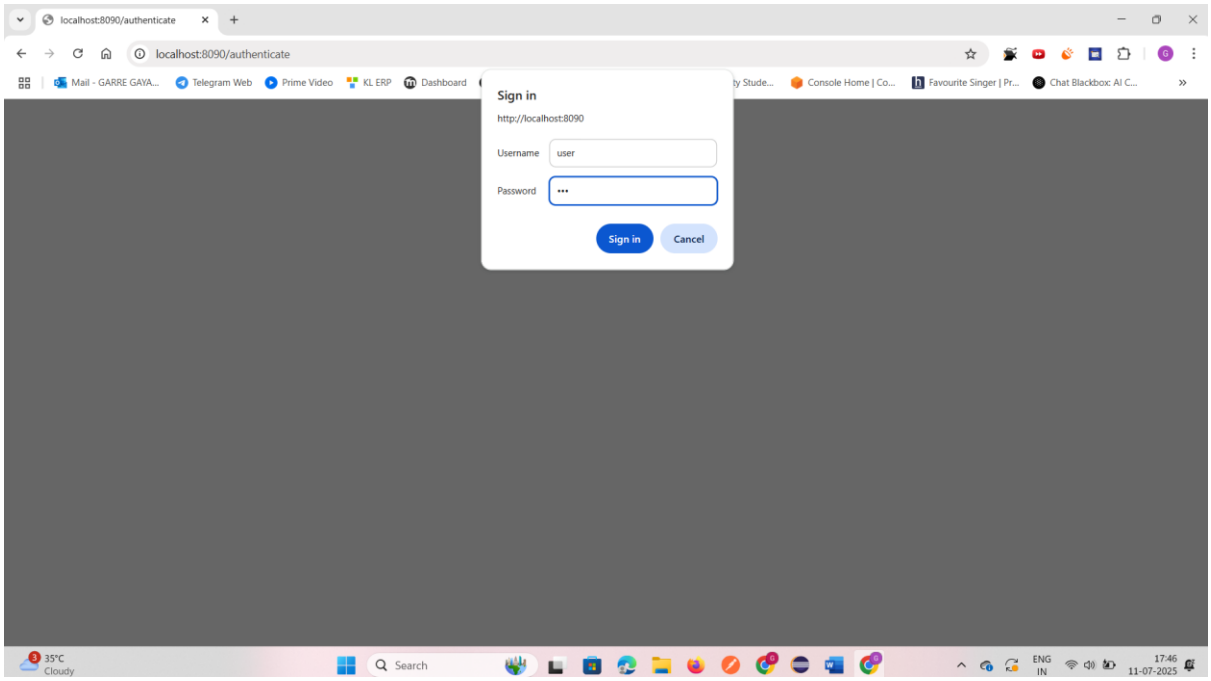
```
        );
    }
    @Bean
    public PasswordEncoder passwordEncoder() {
        return NoOpPasswordEncoder.getInstance(); // Just for demo
    }
}
```

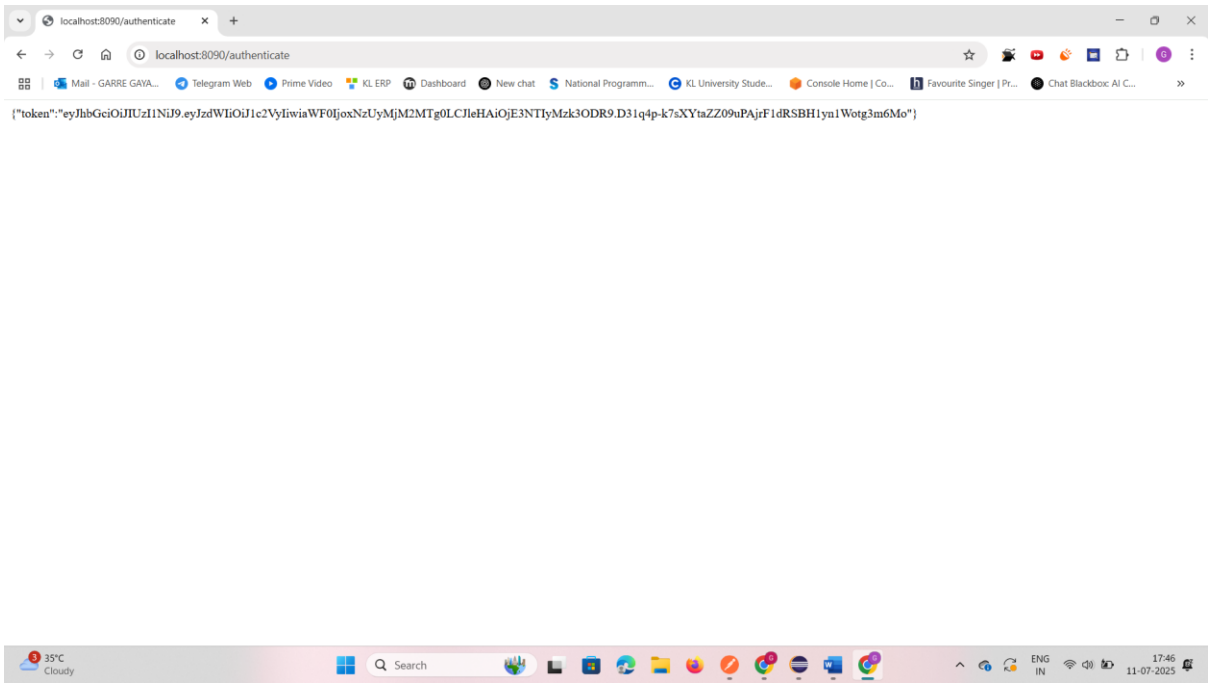
OUTPUTS:




Home
Workspaces
API Network
Search Postman
Invite
Upgrade
http://localhost:8090/authenticate
Save
Share
GET
http://localhost:8090/authenticate
Send
Params
Authorization
Headers (7)
Body
Scripts
Settings
Cookies
Auth Type
Basic Auth
The authorization header will be automatically generated when you send the request. Learn more about Basic Auth authorization.
Username
user
Password
Body
Cookies
Headers (11)
Test Results
200 OK
9 ms
476 B
Raw
Preview
Visualize
{"token":"eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyIiwiaWF0IjoxNzUyMjM1ODUyLCJleHAiOjE3NTIyMzk0NTJ9.Ge4GZtMha9z3mGYGej1sTEmAWL-tp7nolpJhWLIsd2c"}
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash
35°C
Cloudy
Search
ENG
IN
17:45
11-07-2025


localhost:8090/authenticate
Sign in
http://localhost:8090
Username
Password
Sign in
Cancel
35°C
Cloudy
Search
ENG
IN
17:46
11-07-2025


localhost:8090/authenticate
Sign in
http://localhost:8090
Username
user
Password
Sign in
Cancel


localhost:8090/authenticate
{"token":"eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyIiwiaWF0IjoxNzUyMjM2MTg0LCJleHAiOjE3NTIyMzk3ODR9.D31q4p-k7sXYtaZZ09uPAjrF1dRSBH1yn1Wotg3m6Mo"}