**SpringBoot Adding Spring Security Login, AddingRules and BcryptPassword**

**WebSecurityConfig.java**

package org.studyeasy.SpringStarter.config;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.config.Customizer;

import org.springframework.security.config.annotation.method.configuration.EnableMethodSecurity;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

import org.springframework.security.crypto.password.PasswordEncoder;

import org.springframework.security.web.SecurityFilterChain;

@Configuration

@EnableWebSecurity

@EnableMethodSecurity(prePostEnabled = true, securedEnabled = true)

public class WebSecurityConfig {

        private static final String[] WHITELIST = {

                        "/",

                        "/login",

                        "/register",

                        "/db-console/\*\*",

                        // Allow all the css and other resources

                        "/css/\*\*",

                        "/fonts/\*\*",

                        "/images/\*\*",

                        "/js/\*\*"

        };

        @Bean

        public static PasswordEncoder passwordEncoder() {

                return new BCryptPasswordEncoder();

        }

        @Bean

        public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

                http

                  .authorizeHttpRequests(auth -> auth

                  .requestMatchers(WHITELIST)

                  .permitAll()

                  .anyRequest()

                 .authenticated())

                 .formLogin(formLogin -> formLogin

                     .loginPage("/login")

                     .loginProcessingUrl("/login")

                     .usernameParameter("email")

                     .passwordParameter("password")

                     .defaultSuccessUrl("/", true)

                     .failureUrl("/login?error")

                     .permitAll())

                 .logout(logout -> logout

                     .logoutUrl("/logout")

                     .logoutSuccessUrl("/logout?success")

                     .permitAll())

// Enable HTTP Basic authentication

                     .httpBasic(Customizer.withDefaults());

                // TODO: remove these files after upgrading the DB from h2

// infile DB

                http.csrf(csrf -> csrf.disable());

                // Disable frame options

                http.headers(headers -> headers.frameOptions(frameOptions -> frameOptions.disable()));

                return http.build();

        }

}

**AccountService.java -> Updated**

public class AccountService {

    @Autowired

    private AccountRepository accountRepository;

    @Autowired

    private PasswordEncoder passwordEncoder;

    public Account save(Account account){

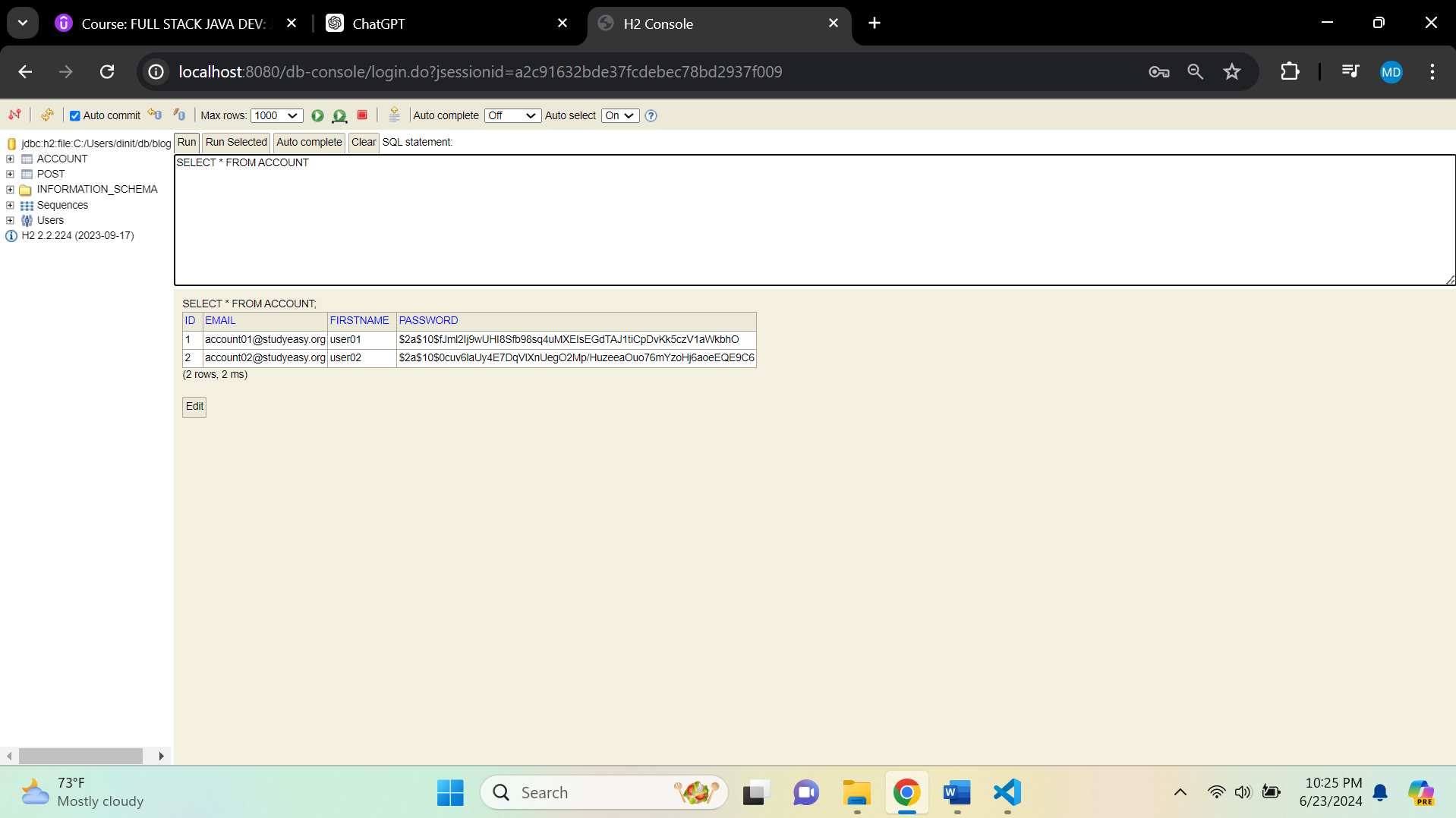
        account.setPassword(passwordEncoder.encode(account.getPassword()));

        return accountRepository.save(account);

    }

}

**Output:-**

****