**SpringBoot\_Email Service Password Change UI**

**Add this in application.properties**

# Site domain - base url -  email password settings

site.domain = http://localhost:8080/

**Add this in AccountController.java**

 @Value("${site.domain}")

 private String site\_domain;

**Update From below code**

String reset\_message = "This is the reset password link : http://localhost/reset-password?token"+ reset\_token;

**To**

String reset\_message = "This is the reset password link : "+site\_domain+"change password?token ="+ reset\_token;

**Adding endpoint for change password in AccountController.java – Email**

// Token is the parameter which is present in the URL, that was sent when reset

    // button clicked and stored in db ; ("token")

    @GetMapping("/change-password")

    public String change\_password(Model model, @RequestParam("token") String token, RedirectAttributes attributes) {

        Optional<Account> optional\_account = accountService.findByToken(token);

        if (optional\_account.isPresent()) {

            long account\_id = optional\_account.get().getId();

            LocalDateTime now = LocalDateTime.now();

            if (now.isAfter(optional\_account.get().getPassword\_reset\_token\_expiry())) {

                attributes.addFlashAttribute("error", "Token expired");

                return "redirect:/forgot-password";

            }

            model.addAttribute("account\_id", account\_id);

            return "account\_views/change-password";

        }

        attributes.addFlashAttribute("error", "Invalid token");

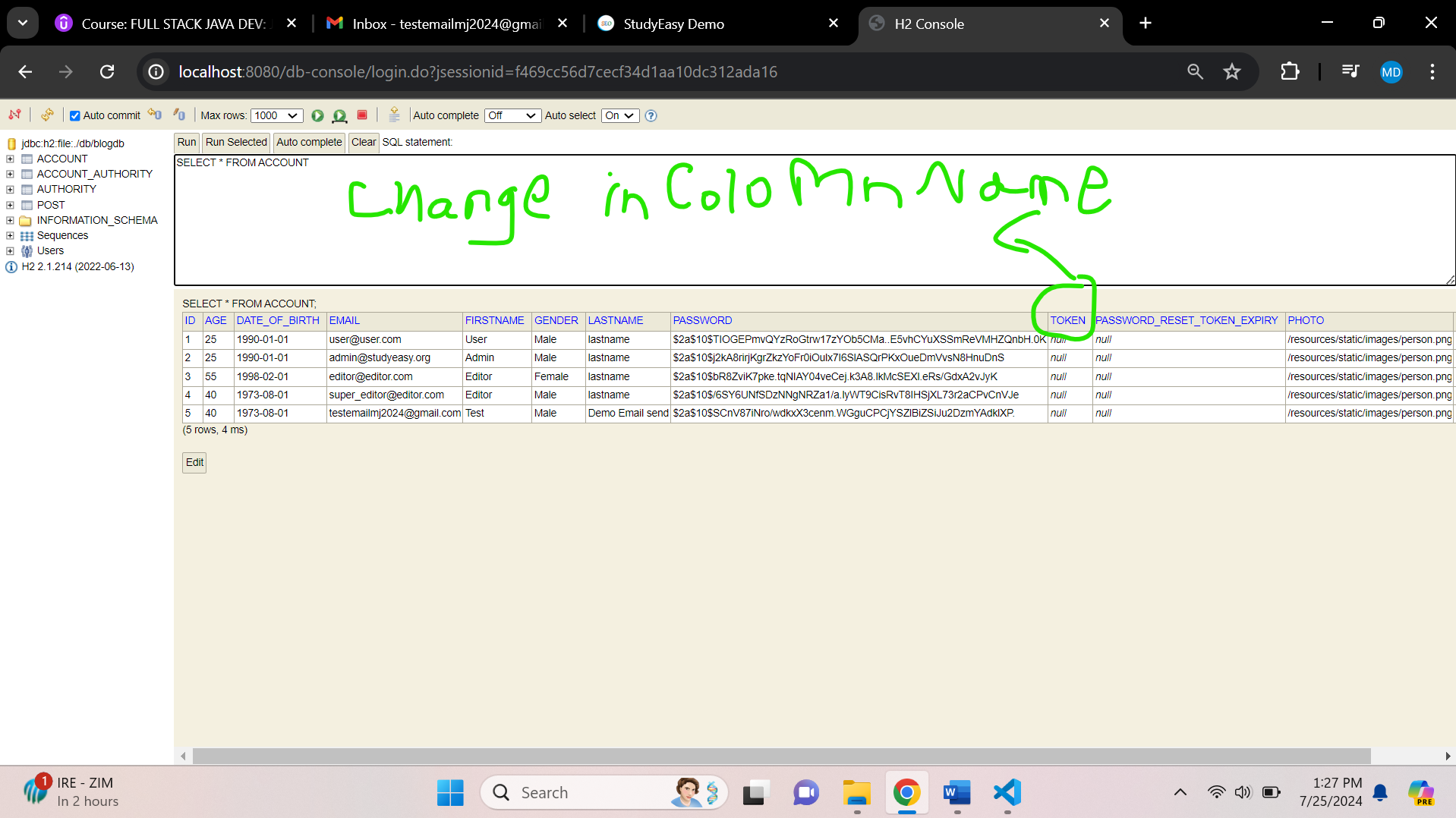
        return "redirect:/forgot-password";

    }

**In Account model -> modify the property -> password\_reset\_token**

@Column(name = "token")

private String password\_reset\_token;

****

**In AccountRepository -> add this code below**

 public Optional<Account> findByToken(String token) {

  return accountRepository.findByToken(token);

}

**In AccountService -> add this code below**

public Optional<Account> findByToken(String token) {

   return accountRepository.findByToken(token);

}

**In Account model -> modify the property -> password\_reset\_token**

**But as we have put the column as token -> change it to token**

@Column(name = "token")

private String token;

**In reset-password.html -> Update like below**

 @PostMapping("/reset-password")

    public String reset\_password(@RequestParam("email") String \_email, RedirectAttributes attributes, Model model) {

        Optional<Account> optional\_account = accountService.findOneByEmail(\_email);

        if (optional\_account.isPresent()) {

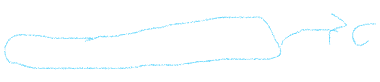
            Account account = accountService.findById(optional\_account.get().getId()).get();

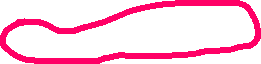
            // UUID Token Generation

            String reset\_token = UUID.randomUUID().toString();

            // Setters



            account.setToken(reset\_token);



            account.setPassword\_reset\_token\_expiry(LocalDateTime.now().plusMinutes(password\_token\_timeout));

            // Saving details

            accountService.save(account);

            // Email Code

            String reset\_message = "This is the reset password link: " + site\_domain + "change-password?token="

                    + reset\_token;

            emailDetails emailDetails = new emailDetails(account.getEmail(), reset\_message,

                    "Reset Password Studyeasy demo");

            if (emailService.sendSimpleEmail(emailDetails) == false) {

                attributes.addFlashAttribute("error", "Error while sending email, contact admin");

            }

            // Final Flash message

            attributes.addFlashAttribute("message", "Password reset email sent");

            return "redirect:/login";

        } else {

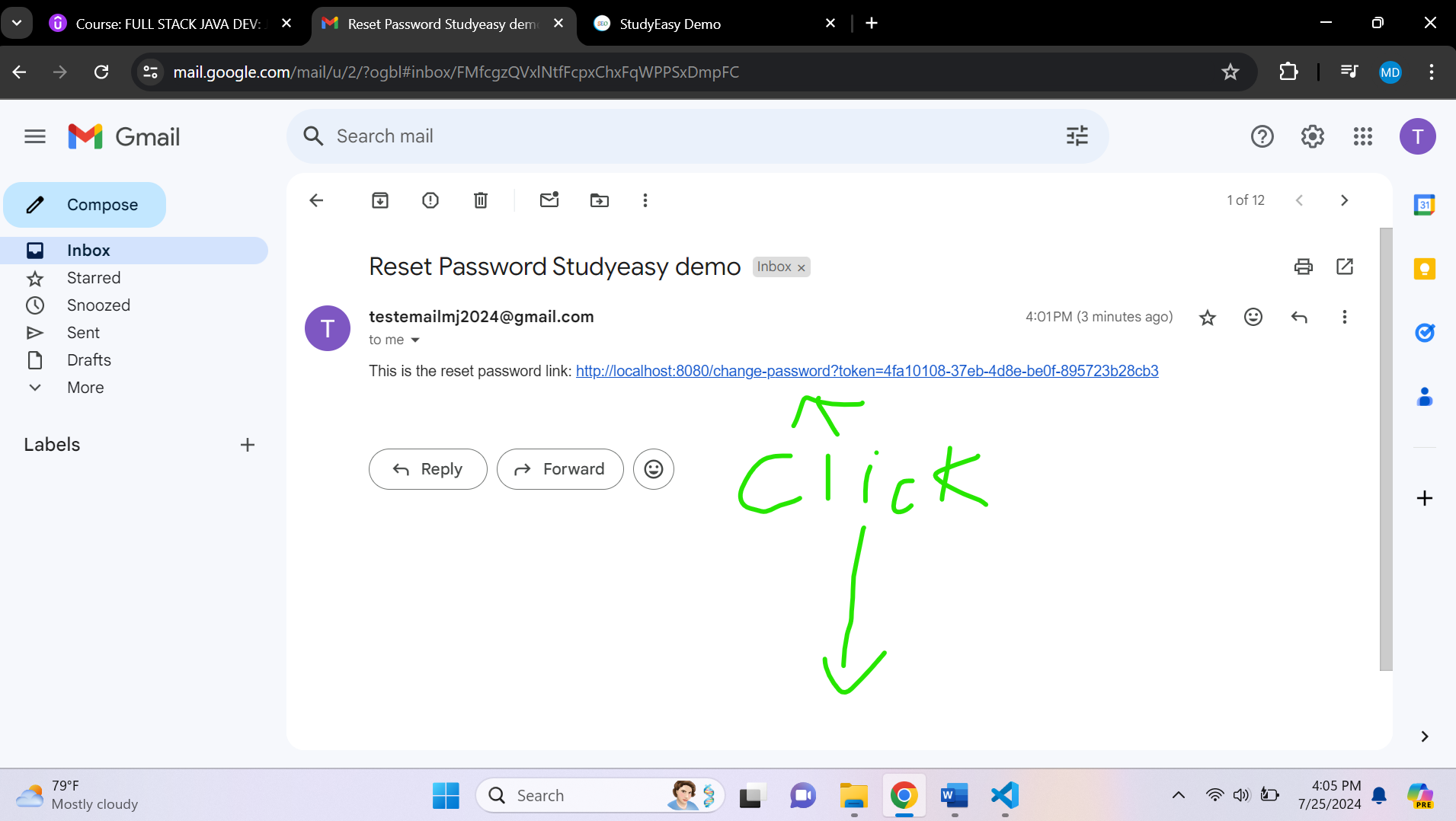
            attributes.addFlashAttribute("error", "No user found with the email supplied");

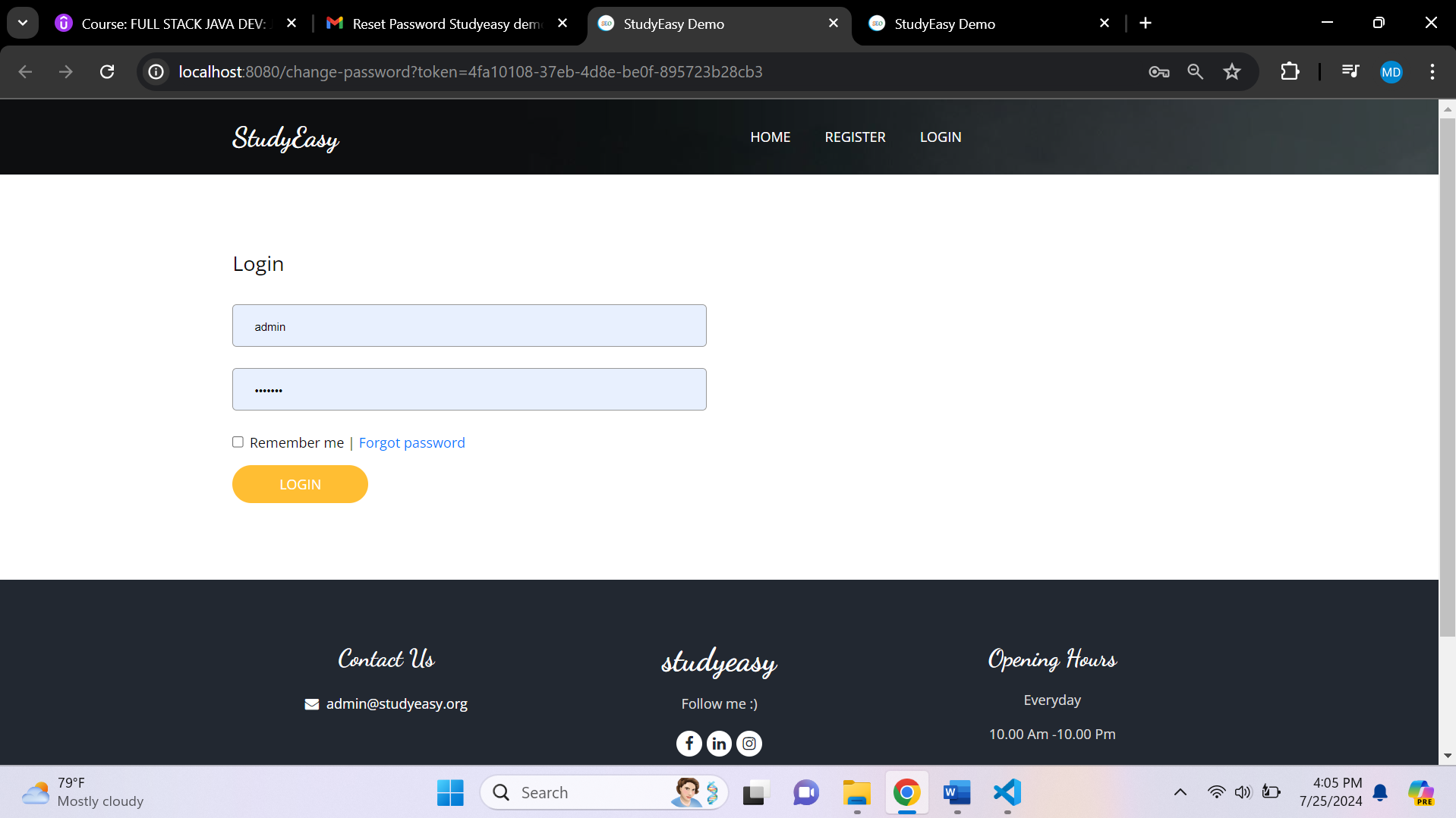
            return "redirect:/forgot-password";

        }

    }

**Output**

****

****