**SpringBoot - Update Profile on the Spring Boot Blog**

**Create a folder -> in the -> src -> resources -> uploads**

**Attach a new dependency -> Apache commons**

   <!-- https://mvnrepository.com/artifact/org.apache.commons/commons-lang3 -->

   <dependency>

     <groupId>org.apache.commons</groupId>

     <artifactId>commons-lang3</artifactId>

     <version>3.15.0</version>

   </dependency>

**In profile.html -> add below code -> Add Image Input file to update**

<!--Adding Image File Input Form-->

 <form method="post" th:action="@{/update\_photo}" enctype="multipart/form-data">

    <div class="form-group">

        <input type="file" name="file" class="form-control-file">

     </div>

 <button type="submit" class="btn btn-primary">Update photo</button></form>

**In below code**

 <!--Adding Photo in this code-->

 <img th:src="@{${photo}}" class="rounded-circle" style="width: 150px; float: left; padding: 10px 10px 10px 10px;" alt="Avatar"  />

 <!--Adding Image File Input Form-->

 <form method="post" th:action="@{/update\_photo}" enctype="multipart/form-data">

    <div class="form-group">

        <input type="file" name="file" class="form-control-file">

     </div>

 <button type="submit" class="btn btn-primary">Update photo</button>

**In AccountController -> add below code**

 @PostMapping("/update\_photo")

    @PreAuthorize("isAuthenticated()")

    public String update\_photo(@RequestParam("file") MultipartFile file, RedirectAttributes attributes, Principal principal) {

        if(file.isEmpty()){

            attributes.addFlashAttribute("error", "No file uploaded");

            return "redirect:/profile";

        }

        return "";

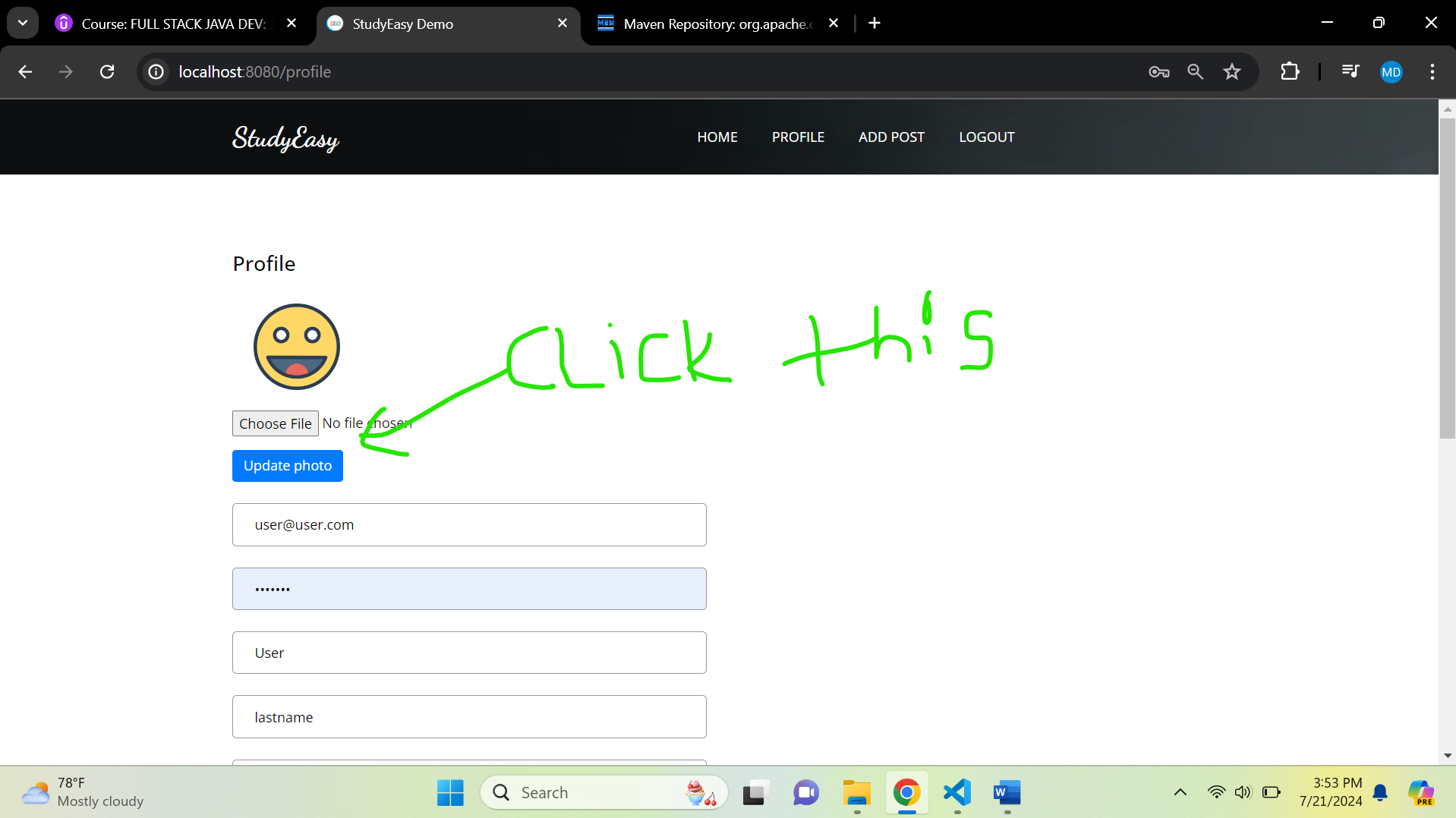
    }

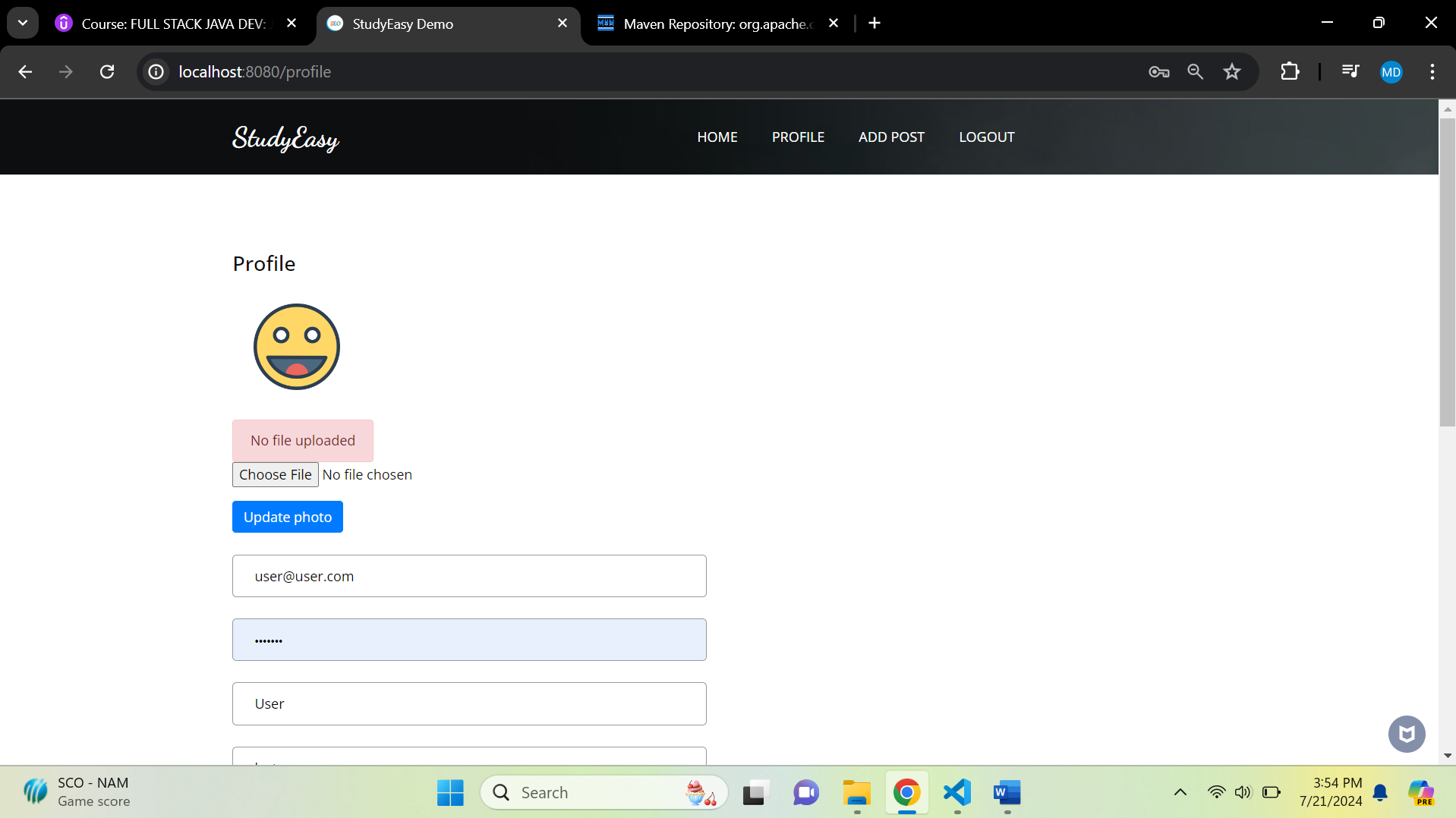
**In profile.html -> A flash message to be added**

 <!--Paragraph flash message-->

      <p th:text="${error}" th:if="${error ne null}" class="alert alert-danger"></p>

**Output**

****

****

**Create a AppUtil.java in utils folder**

package org.studyeasy.SpringBlog.util;

import java.io.File;

public class AppUtil {

    public static String get\_upload\_path(String fileName){

        return new File("src\\main\\resources\\static\\uploads").getAbsolutePath() + "\\" + fileName;

    }

}

**Using \\ is very important as it is also escape sequence – not one single slash.**

**@Value only works if it is controller or service layer.**

**In reference to :**

# Static files setting

spring.mvc.static-path-pattern=/resources/static/\*\*

  @Value("${spring.mvc.static-path-pattern}")

  private String photo\_prefix;

**In AccountController -> Add below code**

   @PostMapping("/update\_photo")

    @PreAuthorize("isAuthenticated()")

    public String update\_photo(@RequestParam("file") MultipartFile file, RedirectAttributes attributes, Principal principal) {

        if(file.isEmpty()){

            attributes.addFlashAttribute("error", "No file uploaded");

            return "redirect:/profile";

        }else{

            String fileName = StringUtils.cleanPath(file.getOriginalFilename());

            try{

                int length = 10;

                boolean useLetters = true;

                boolean useNumbers = true;

                String generatedString = RandomStringUtils.random(length, useLetters, useNumbers);

                String final\_photo\_name = generatedString + fileName;

                String absolute\_fileLocation = AppUtil.get\_upload\_path(final\_photo\_name);

                Path path = Paths.get(absolute\_fileLocation);

                Files.copy(file.getInputStream(), path, StandardCopyOption.REPLACE\_EXISTING);

                attributes.addFlashAttribute("message", "You successfully uploaded");

                String authUser = "email";

                if (principal != null) {

                    authUser = principal.getName();

                }

                Optional<Account> optionalAccount = accountService.findOneByEmail(authUser);

                if (optionalAccount.isPresent()) {

                    Account account = optionalAccount.get();

                    Account account\_by\_id = accountService.findOneById(account.getId()).get();

                    String relative\_fileLocation = photo\_prefix.replace("\*\*", "uploads/"+final\_photo\_name );

                    account\_by\_id.setPhoto(relative\_fileLocation);

                    accountService.save(account\_by\_id);

                }

                try{

                    TimeUnit.SECONDS.sleep(1);

                }catch(InterruptedException ie){

                    Thread.currentThread().interrupt();

                }

                return "redirect:/profile";

            }catch(Exception e){

                //TODO : handle exception

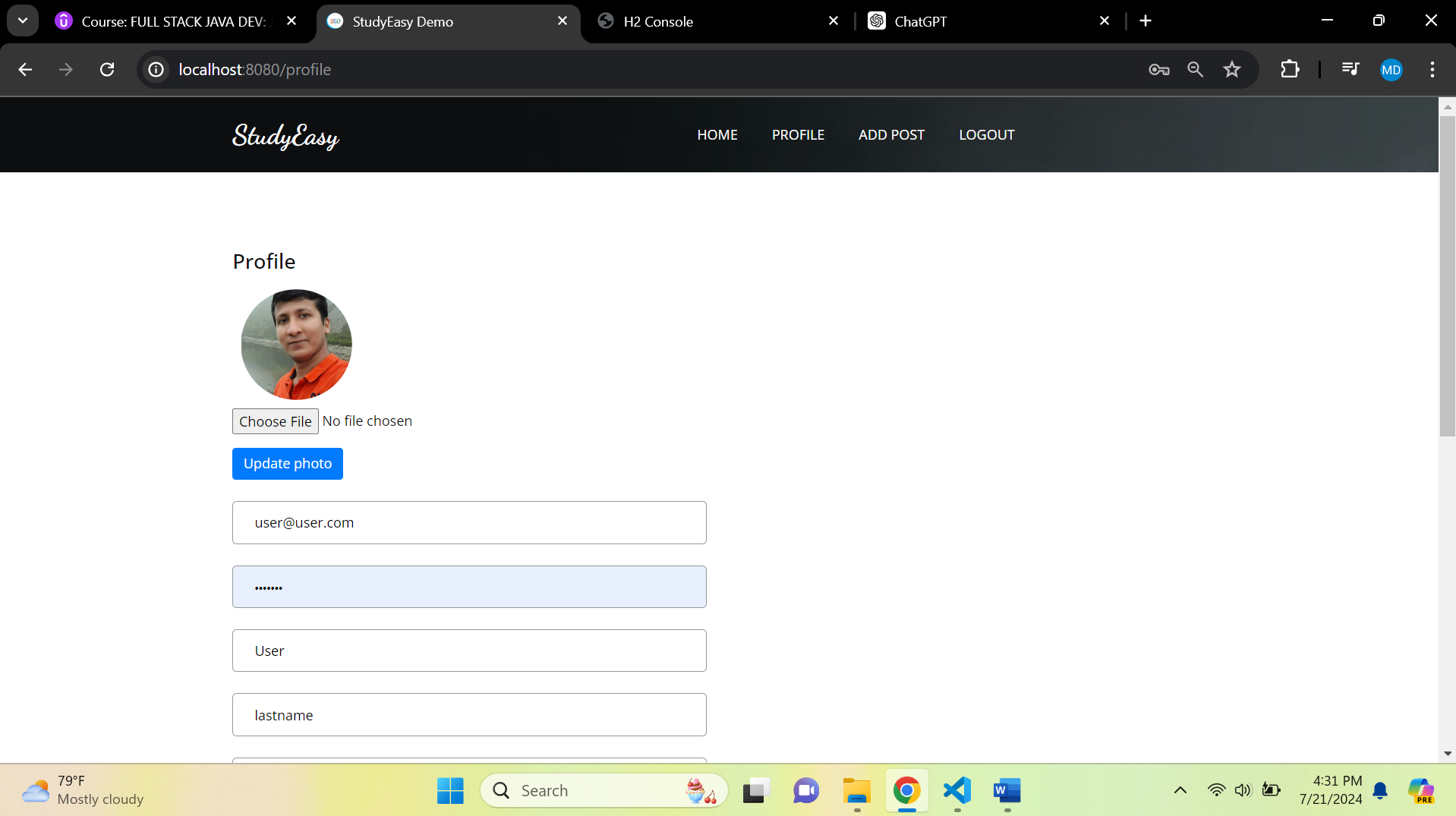
            }

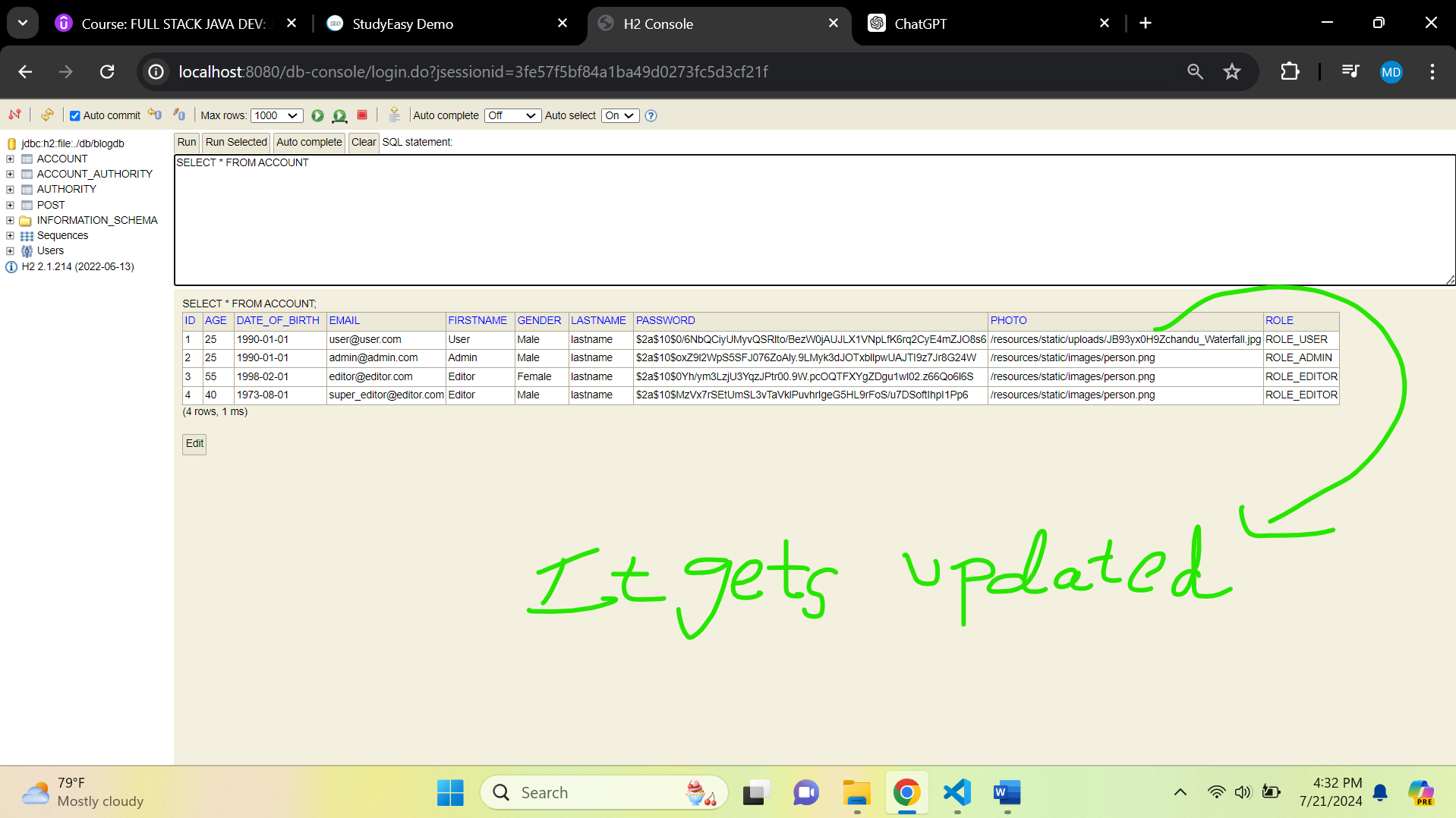
        }

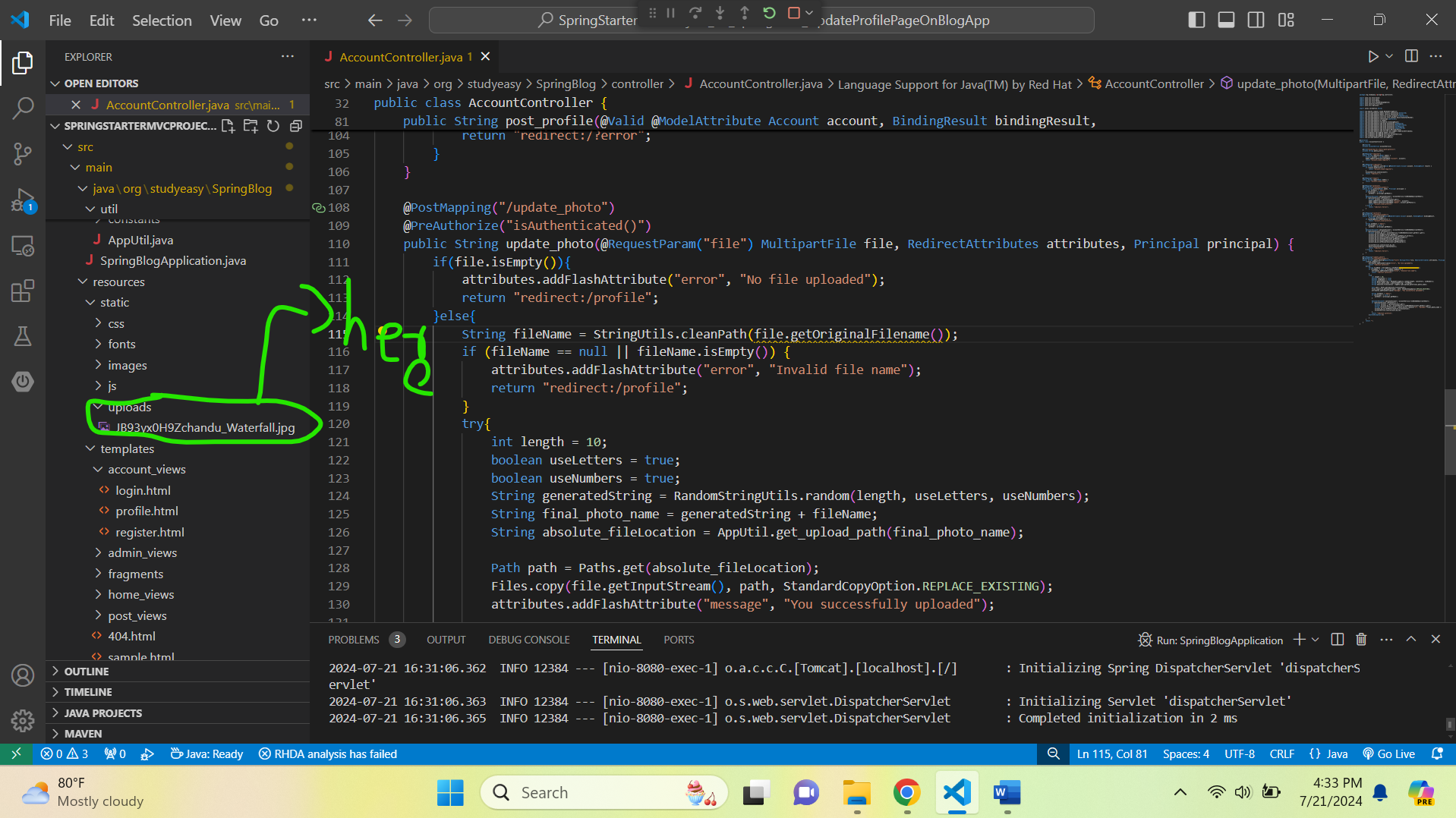
        return "";

    }

**Output**

****

****

****

**Updated AccountController method -> update\_photo**

 @PostMapping("/update\_photo")

    @PreAuthorize("isAuthenticated()")

    public String update\_photo(@RequestParam("file") MultipartFile file, RedirectAttributes attributes,

            Principal principal) {

        if (file.isEmpty()) {

            attributes.addFlashAttribute("error", "No file uploaded");

            return "redirect:/profile";

        } else {

            String fileName = file.getOriginalFilename();

            if (fileName == null || fileName.isEmpty()) {

                attributes.addFlashAttribute("error", "Invalid file name");

                return "redirect:/profile";

            }

            String cleanedFileName = StringUtils.cleanPath(fileName);

            try {

                int length = 10;

                boolean useLetters = true;

                boolean useNumbers = true;

                String generatedString = RandomStringUtils.random(length, useLetters, useNumbers);

                String finalPhotoName = generatedString + cleanedFileName;

                String absoluteFileLocation = AppUtil.get\_upload\_path(finalPhotoName);

                Path path = Paths.get(absoluteFileLocation);

                Files.copy(file.getInputStream(), path, StandardCopyOption.REPLACE\_EXISTING);

                attributes.addFlashAttribute("message", "You successfully uploaded");

                String authUser = "email";

                if (principal != null) {

                    authUser = principal.getName();

                }

                Optional<Account> optionalAccount = accountService.findOneByEmail(authUser);

                if (optionalAccount.isPresent()) {

                    Account account = optionalAccount.get();

                    Account accountById = accountService.findOneById(account.getId()).orElse(null);

                    if (accountById != null) {

                        String relativeFileLocation = photo\_prefix.replace("\*\*", "uploads/" + finalPhotoName);

                        accountById.setPhoto(relativeFileLocation);

                        accountService.save(accountById);

                    }

                    try {

                        TimeUnit.SECONDS.sleep(1);

                    } catch (InterruptedException ie) {

                        Thread.currentThread().interrupt();

                    }

                }

                return "redirect:/profile";

            } catch (Exception e) {

                attributes.addFlashAttribute("error", "An error occurred while uploading the photo");

                return "redirect:/profile";

            }

        }

    }

**The error was due not proper closing of form tag- </form> – ErrorLog – 21/7/2024**

**so, when I try to click on update data the validation for photo should not come in between**