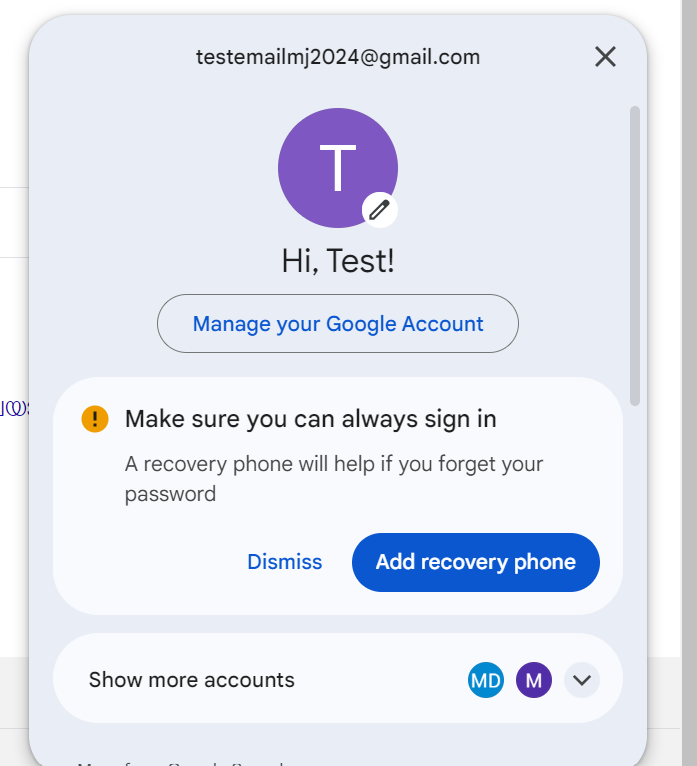
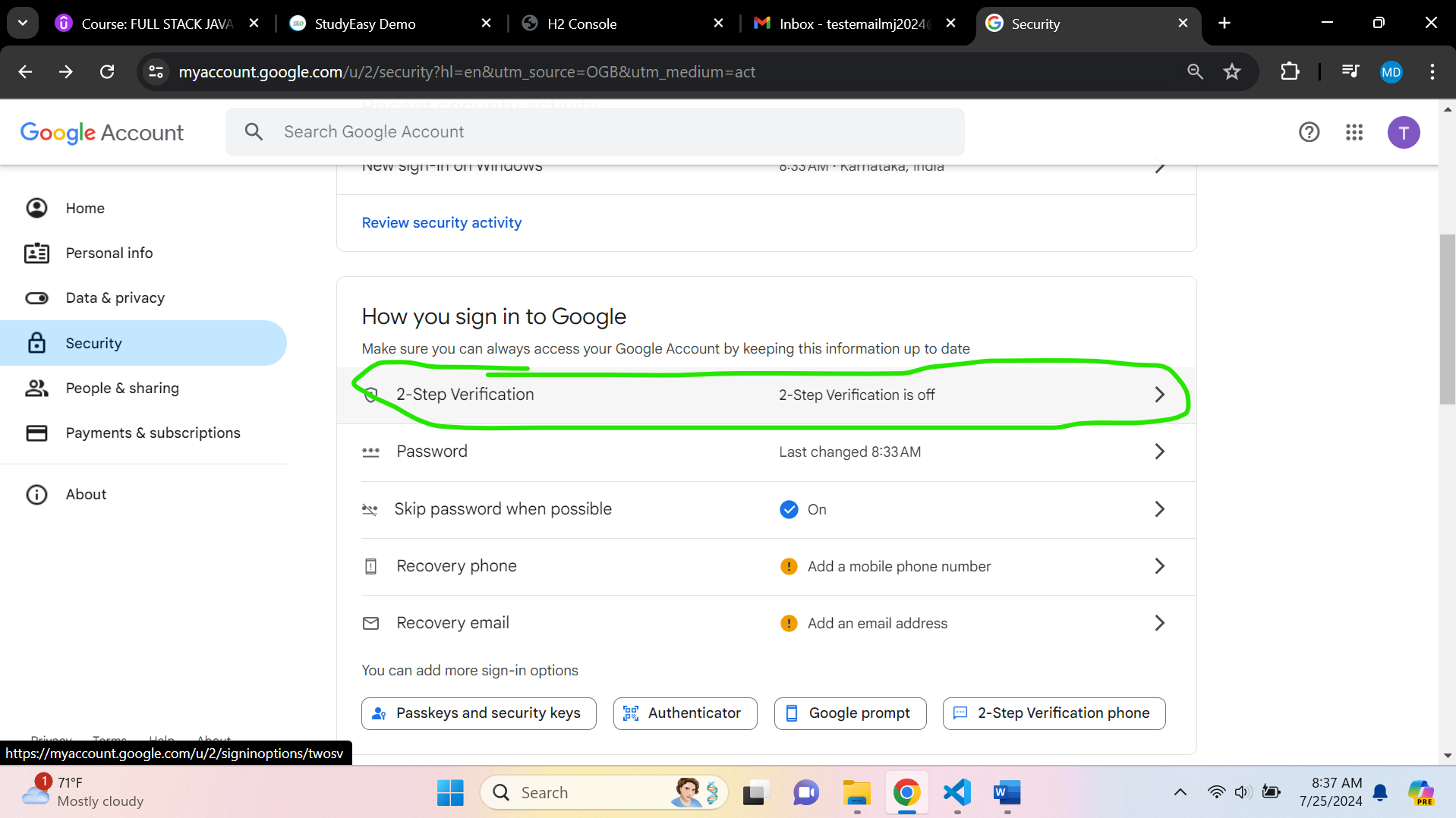
**SpringBoot – Sending Email for Password Reset - Continues**

When it comes to Gmail, you have to create App Password in order to send email using application.

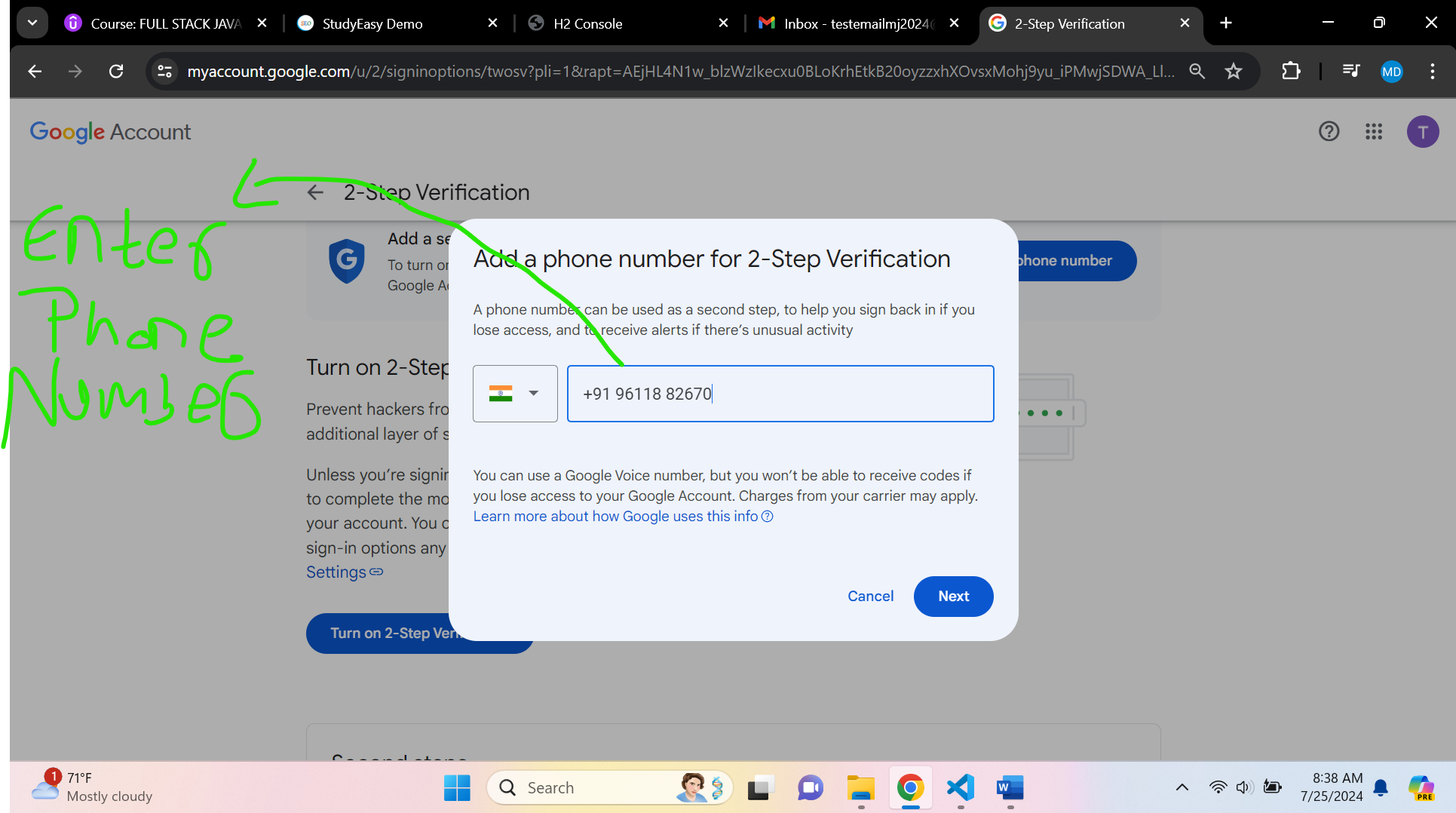
Created a new account under like below



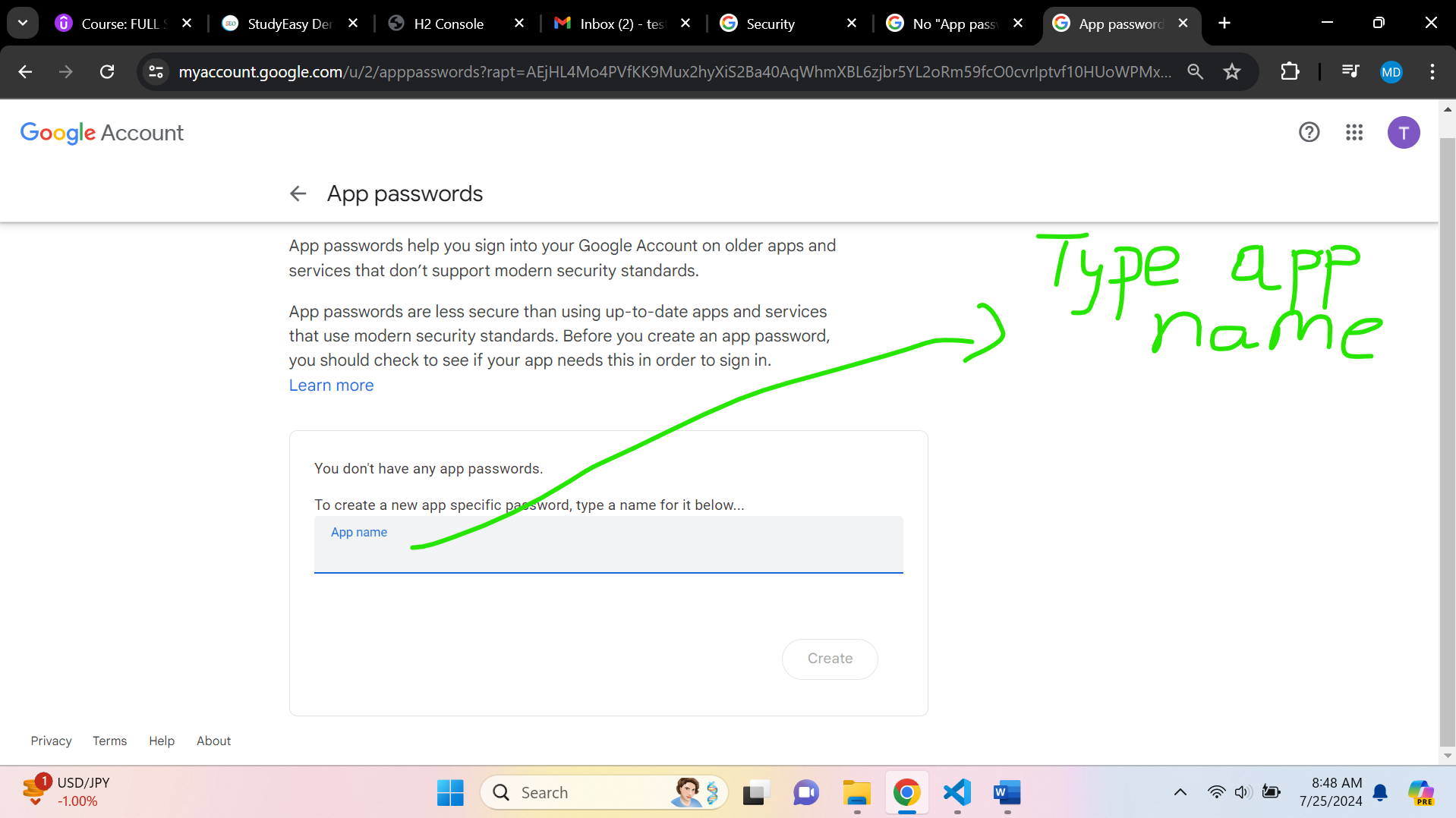
Manage Your Google Account -> Security -> To enable AppPassword -> You have to enable 2 step verification -

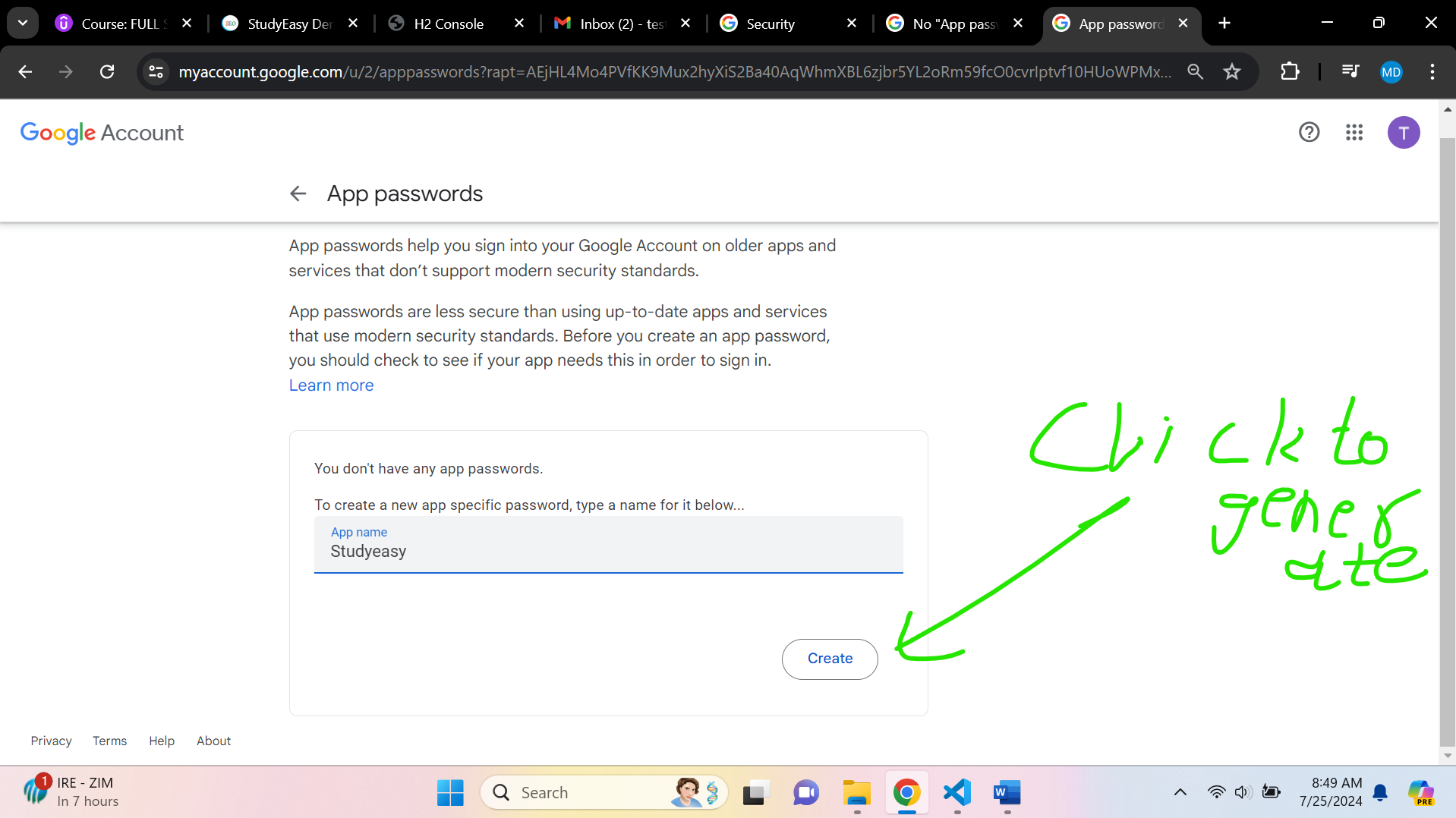


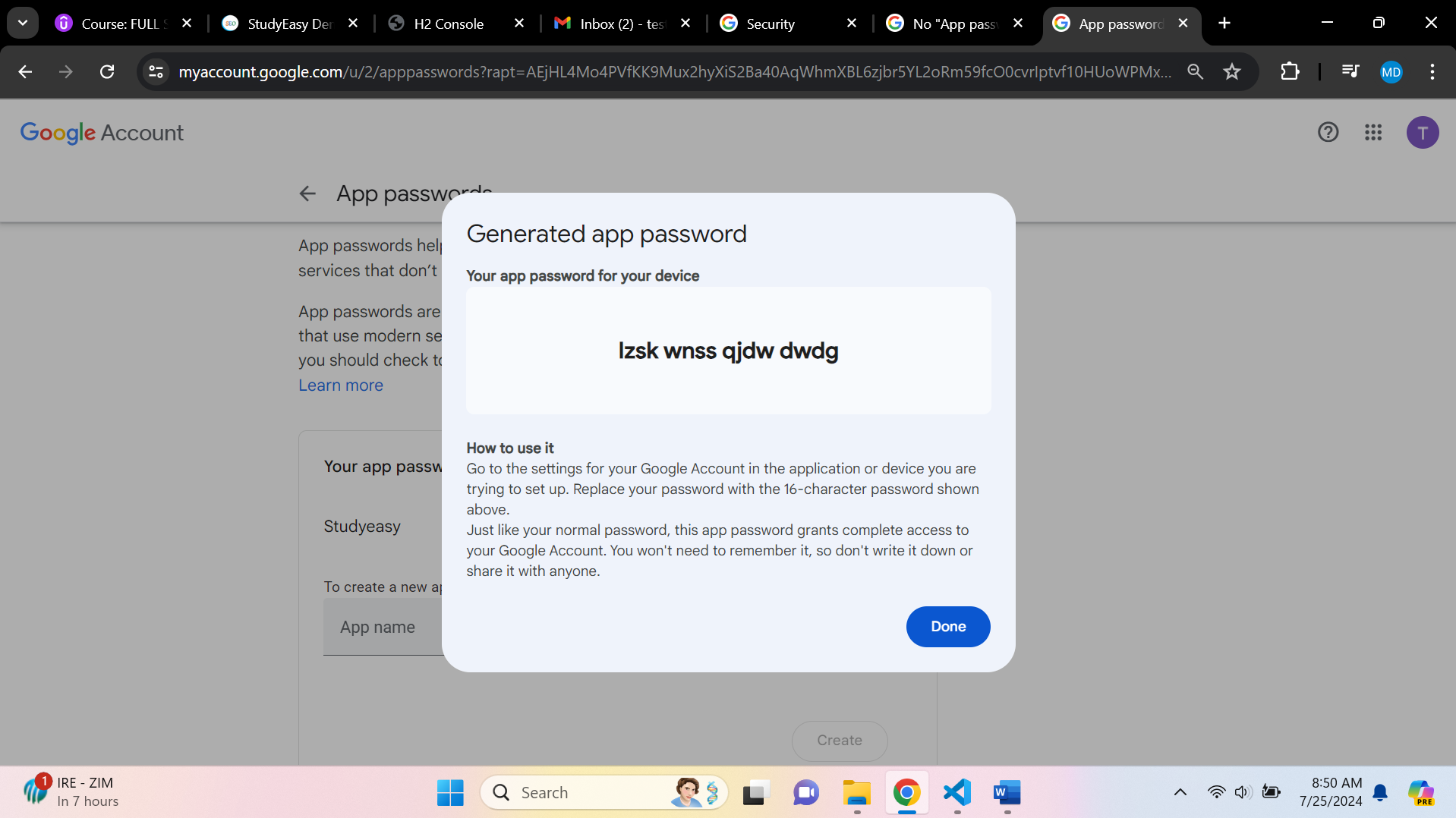
Enter Phone Number



Enter Code and click next







It is a One time Show password. -> **lzsk wnss qjdw dwdg**

**In application.properties -> add below code**

# Email settings

spring.mail.username=testemailmj2024@gmail.com

spring.mail.password=testmailmj2024\_$

spring.mail.host=smtp.gmail.com

spring.mail.port=587

spring.mail.properties.mail.smtp.auth=true

spring.mail.properties.mail.smtp.starttls.enable=true

spring.mail.smtp.ssl.trust=smtp.gmail.com

spring.mail.transport.protocol=smtp

Create a new file -> secret.properties in resources folder -> Move it to new file

secret.properties

spring.mail.username=testemailmj2024@gmail.com

spring.mail.password=testmailmj2024\_$

Import this file

# Spring Import - secret.properties

spring.config.import=secret.properties

or

# Spring Import - secret.properties

# spring.config.import=secret.properties

# Spring Import - secret.properties - make this import as optional

spring.config.import=optional:secret.properties

-------------------------------------Part 2----------------------------------------------------------------

Add below dependency

<!--Starter Mail -->

<dependency>

   <groupId>org.springframework.boot</groupId>

   <artifactId>spring-boot-starter-mail</artifactId>

</dependency>

Create a AppConfig in config folder

package org.studyeasy.SpringBlog.config;

import java.util.Properties;

import org.springframework.beans.factory.annotation.Value;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.mail.javamail.JavaMailSender;

import org.springframework.mail.javamail.JavaMailSenderImpl;

@Configuration

public class AppConfig {

    @Value("${mail.transport.protocol}")

    private String mail\_transport\_protocol;

    @Value("${spring.mail.host}")

    private String spring\_mail\_host;

    @Value("${spring.mail.port}")

    private String spring\_mail\_port;

    @Value("${spring.mail.username}")

    private String spring\_mail\_username;

    @Value("${spring.mail.password}")

    private String spring\_mail\_password;

    @Value("${spring.mail.properties.mail.smtp.auth}")

    private String mail\_smtp\_auth;

    @Value("${spring.mail.properties.mail.smtp.starttls.enable}")

    private String mail\_smtp\_starttls\_enable;

    @Value("${spring.mail.smtp.ssl.trust}")

    private String smtp\_ssl\_trust;

    @Bean

    public JavaMailSender  gtJavaMailSender(){

        JavaMailSenderImpl mailSender = new JavaMailSenderImpl();

        mailSender.setHost(spring\_mail\_host);

        mailSender.setPort(Integer.parseInt(spring\_mail\_port));

        mailSender.setUsername(spring\_mail\_username);

        mailSender.setPassword(spring\_mail\_password);

        Properties props = mailSender.getJavaMailProperties();

        //put(name of the property, value of property)

        props.put(mailSender, props);

        props.put("mail.transport.protocol", mail\_transport\_protocol);

        props.put("mail.smtp.auth", mail\_smtp\_auth);

        props.put("mail.smtp.starttls.enable", mail\_smtp\_starttls\_enable);

        props.put("mail.debug", "true");

        props.put("mail.smtp.ssl.trust",smtp\_ssl\_trust);

        return mailSender;

    }

}

Create a new file in util folder -> email folder -> emailDetails.java

package org.studyeasy.SpringBlog.util.email;

import lombok.Data;

@Data

public class emailDetails {

    private String recipient;

    private String msgBody;

    private String subject;

}

Create a new file in service folder -> EmailService.java -> Service Layer for email

package org.studyeasy.SpringBlog.services;

public class EmailService {

}