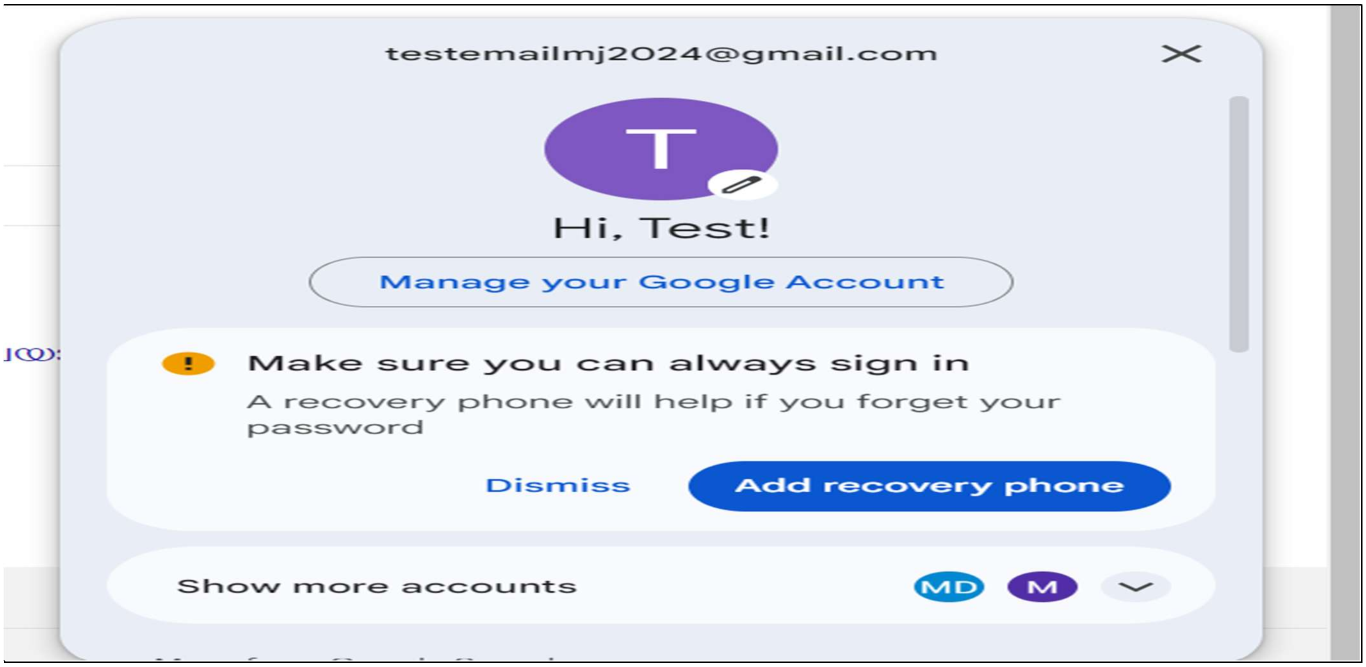
SpringBoot – Sending Email for Password Reset - Continues

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



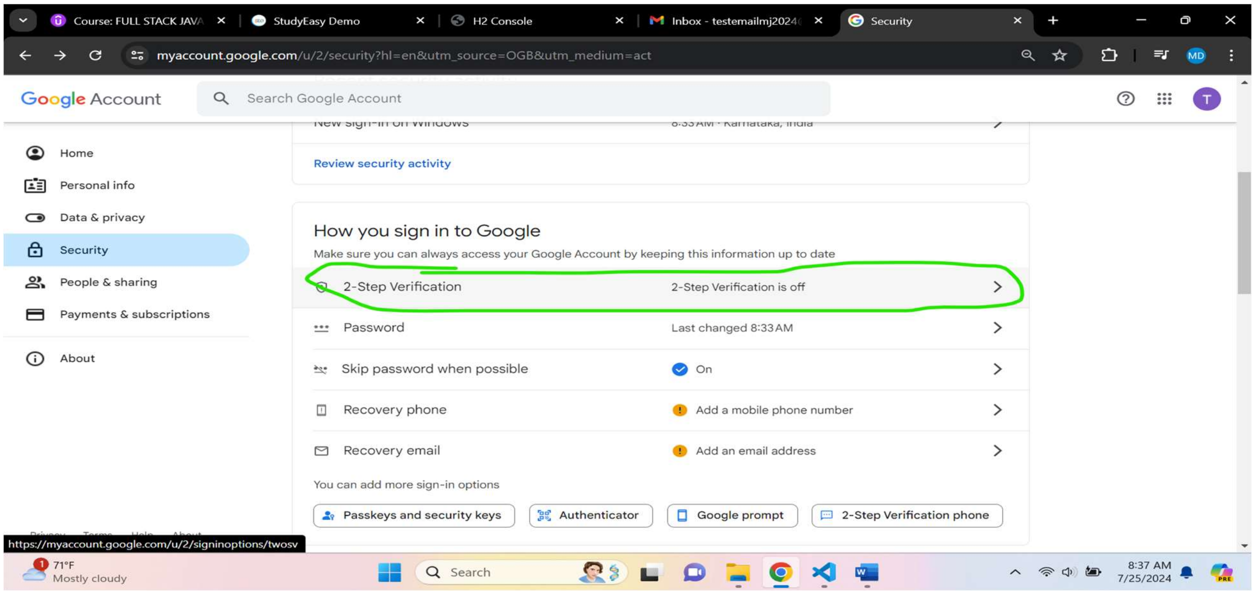
testemailmj2024@gmail.com

Hi, Test!

Make sure you can always sign in

A recovery phone will help if you forget your password

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



How you sign in to Google

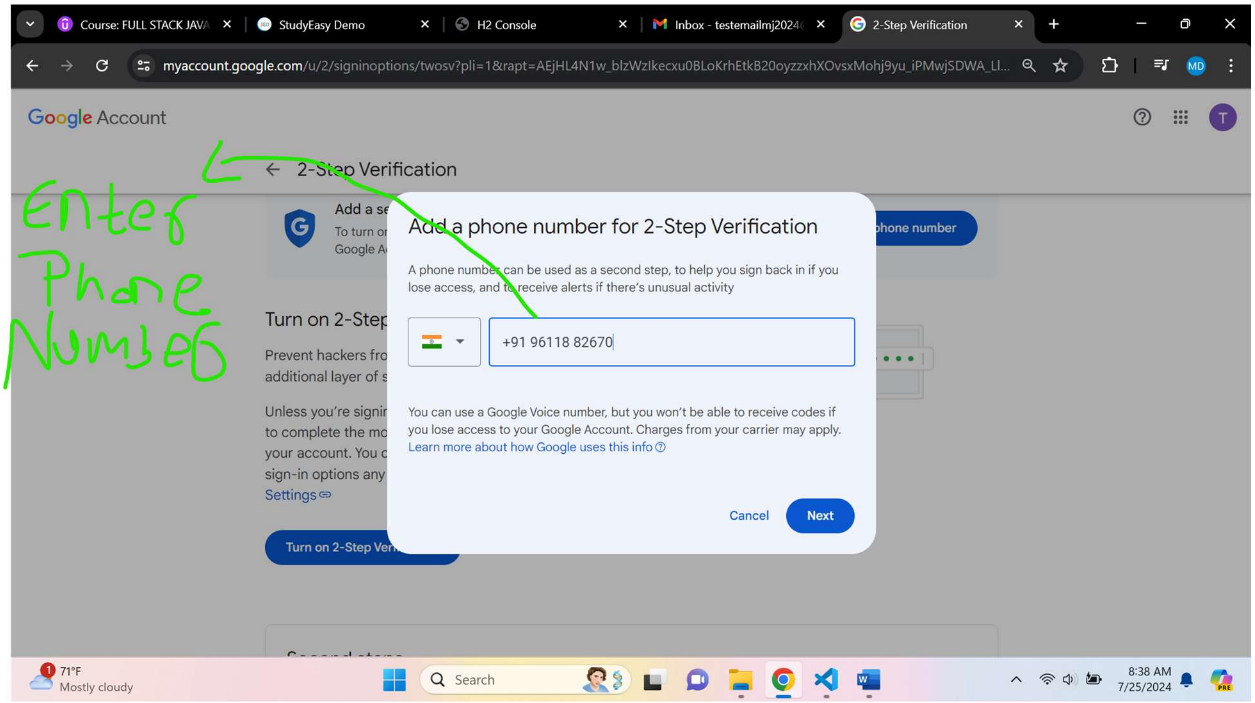
Make sure you can always access your Google Account by keeping this information up to date

2-Step Verification	2-Step Verification is off
Password	Last changed 8:33 AM
Skip password when possible	On
Recovery phone	Add a mobile phone number
Recovery email	Add an email address

You can add more sign-in options

Passkeys and security keys
Authenticator
Google prompt
2-Step Verification phone
https://myaccount.google.com/u/2/signinoptions/twosv
71°F
Mostly cloudy
Search
8:37 AM
7/25/2024

Enter Phone Number



Add a phone number for 2-Step Verification

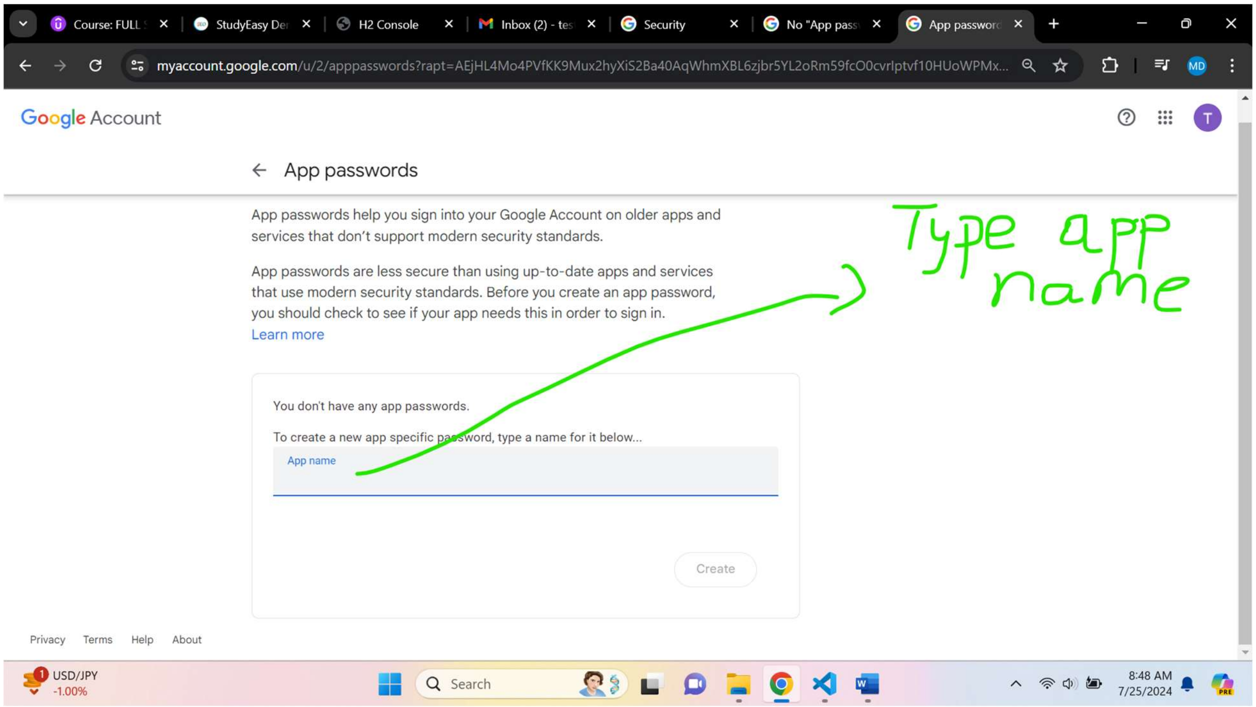
A phone number can be used as a second step, to help you sign back in if you lose access, and to receive alerts if there's unusual activity

+91 96118 82670

You can use a Google Voice number, but you won't be able to receive codes if you lose access to your Google Account. Charges from your carrier may apply. Learn more about how Google uses this info

Cancel Next

Enter Code and click next



App passwords

App passwords help you sign into your Google Account on older apps and services that don't support modern security standards.

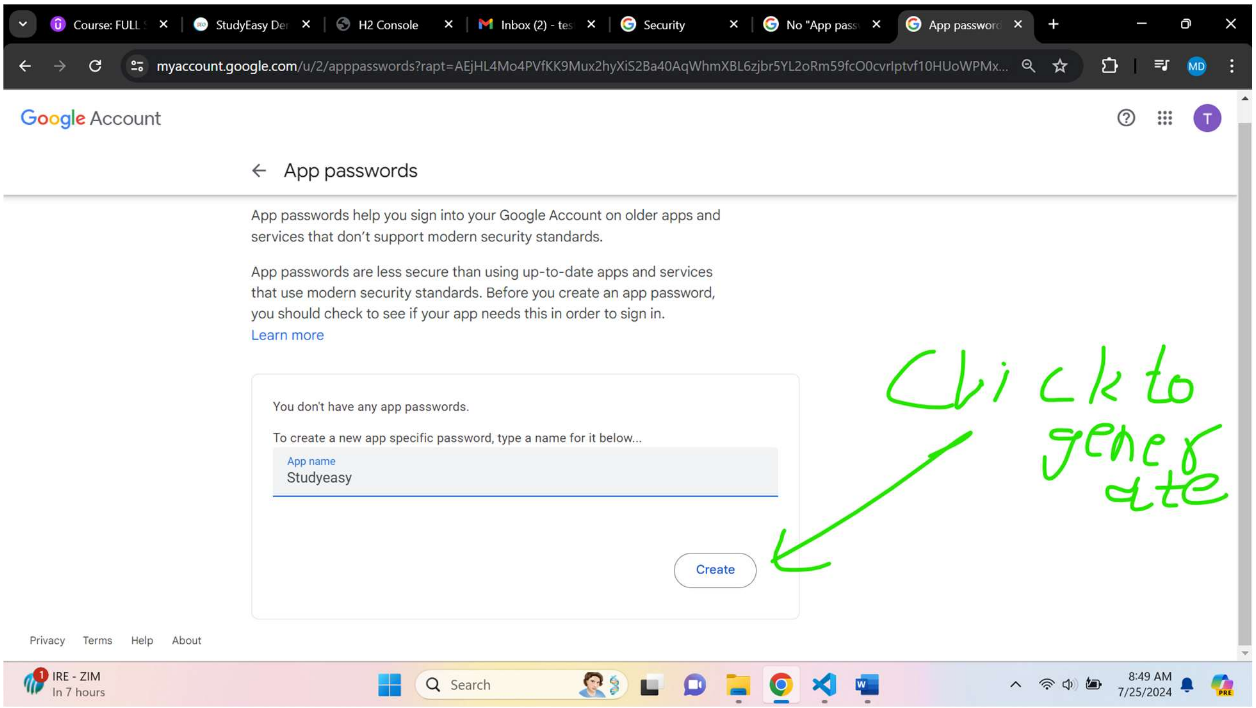
App passwords are less secure than using up-to-date apps and services that use modern security standards. Before you create an app password, you should check to see if your app needs this in order to sign in.
Learn more

You don't have any app passwords.

To create a new app specific password, type a name for it below...

App name

Create



App passwords

App passwords help you sign into your Google Account on older apps and services that don't support modern security standards.

App passwords are less secure than using up-to-date apps and services that use modern security standards. Before you create an app password, you should check to see if your app needs this in order to sign in.
Learn more

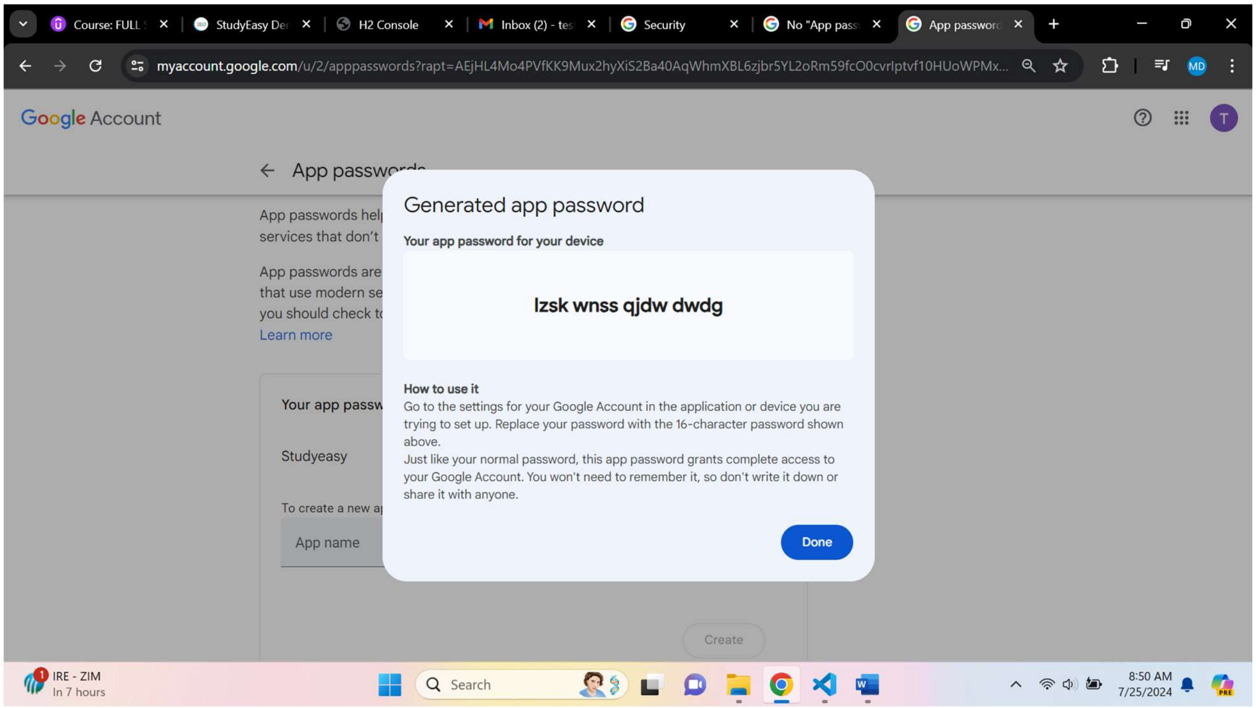
You don't have any app passwords.

To create a new app specific password, type a name for it below...

App name
Studyeasy

Create

Click to generate



Generated app password

Your app password for your device

lzsk wnss qjdw dwdg

How to use it
Go to the settings for your Google Account in the application or device you are trying to set up. Replace your password with the 16-character password shown above.
Just like your normal password, this app password grants complete access to your Google Account. You won't need to remember it, so don't write it down or share it with anyone.

Done

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

```java
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 {

}
```