# SUMMARY

In the digitalized world, studies are carried out in the field of medicine, commercial and many other subjects. These works provide incredible convenience and save time in human life. However digitalization provides convenience not only for human life but also for our animal friends. The main purpose of this application, which is written for pet owners and animal friends, is to provide a good connection between pet owners and veterinarians, and to regularly monitor the vaccinations of pets. Thanks to the application, vaccination will be followed. This is important for animal health and indirectly for the health of everyone who comes into contact with the animal.

Pet owners can frequently ask questions to the veterinarian. Considering the frequent communication between the veterinarian and pet owner, if the vet has many customers, there will be a serious message density. For this reason, a question and answer section was made in the application. In this way, the concentration of the messaging network used by the veterinarian in his private life was prevented.

There is also a commercial element in practice. A section has been written for campaigns related to the products that the vet sells. In this way, users will be informed about the campaigns, while veterinarians will be able to increase their gains in product sales.

**CONTENT**

[SUMMARY I](#_Toc43465063)

[FIGURE LIST III](#_Toc43465064)

[INTRODUCTION 1](#_Toc43465065)

[2.USED TECHNOLOGIES 2](#_Toc43465066)

[2.1. Android Studio 2](#_Toc43465067)

[2.1.1. XML 2](#_Toc43465068)

[2.1.2 JAVA 3](#_Toc43465069)

[2.2 Php 3](#_Toc43465070)

[3.PAGE DESIGNS AND FIGURES 4](#_Toc43465071)

[3.1 Home Fragment 4](#_Toc43465072)

[3.2 Entering Page 5](#_Toc43465073)

[3.3 Register Page 6](#_Toc43465074)

[4.ANDROID STUDIO DETAILS 7](#_Toc43465075)

[4.1 Retrofit 7](#_Toc43465076)

[4.1.1 BaseUrl 9](#_Toc43465077)

[4.1.2 ManagerAll 9](#_Toc43465078)

[4.2 Manifest,Permission and First Interface 10](#_Toc43465079)

[4.3 Libraries 11](#_Toc43465080)

[5.WEB SERVICES 12](#_Toc43465081)

[5.1 Web Service Content 14](#_Toc43465082)

[5.1.1 Settings 14](#_Toc43465083)

[5.1.2 Register Service 15](#_Toc43465084)

[5.1.2 Activate 17](#_Toc43465085)

[5.1.4 Login 17](#_Toc43465086)

[5.1.5 Delete Answer 19](#_Toc43465087)

[RESOURCES 20](#_Toc43465088)

[APPRECIATION 21](#_Toc43465089)

[CV 22](#_Toc43465090)

# FIGURE LIST

[**Figure 1\_Home Fragment** 4](file:///C:\Users\Alpay\Desktop\Yeni%20Microsoft%20Word%20Belgesi.docx#_Toc43465091)

[**Figure 2\_Entering Page** 5](#_Toc43465092)

[**Figure 3\_Register Page** 6](#_Toc43465093)

[**Figure 4\_RestApi** 8](#_Toc43465094)

[**Figure 5\_Base Url** 9](#_Toc43465095)

[**Figure 6\_Manager All** 9](#_Toc43465096)

[**Figure 7\_Manifest** 10](#_Toc43465097)

[**Figure 8\_Libraries** 11](#_Toc43465098)

[**Figure 9\_Control Panel** 12](#_Toc43465099)

[**Figure 10\_File Manager** 13](#_Toc43465100)

[**Figure 11\_Settings** 14](#_Toc43465101)

[**Figure 12\_Register Service** 15](#_Toc43465102)

[**Figure 13\_Register Service 2** 16](#_Toc43465103)

[**Figure 14\_Activate Service** 17](#_Toc43465104)

[**Figure 15\_Login Service** 17](#_Toc43465105)

[**Figure 16\_Login Service 2** 18](#_Toc43465106)

[**Figure 17\_Login Service 3** 18](#_Toc43465107)

[**Figure 18\_Delete Service** 19](#_Toc43465108)

# INTRODUCTION

The project consists of two parts. These are web service and android studio. The interface of any activity was created first and the interfaces created are in XML format.Then backend codes were written. Web service codes were written for the interaction of the current activity on the internet. Finally, the interaction of these codes was provided with the use of robo pojo and retrofit library. This process was repeated for each activity, fragment and adapter. Host service was purchased and database usage was realized with mysql.

The php language was used for web service writing. In Android studio, codes were written in java. Fragments were used because there are many functional situations in the project. Adapters were used to establish the connection between the web service and android. Warning messages determined on the web side were also displayed on the android application with Robo Pojo Modelling.

# 2.USED TECHNOLOGIES

# 2.1. Android Studio

The project was developed in Android Studio.

Android Studio is;

* Gradle-based and flexible

Android Studio has;

* multiple APK printouts
* GitHub integration, C ++ and NDK support
* rich drag-and-drop feature that simplifies screen design
* test tools and frameworks to check the application's performance, usability, interoperability in different versions
* easy and secure APK signing
* google services that easy to implement

### 2.1.1. XML

Interface of the project is prepared in Android Studio using with Xml language.

XML is;

* a markup language such as HTML.
* for designed to describe data.

XML has;

* not predefined tags. So the user must create his own tags.

### 2.1.2 JAVA

The remaining code of the project is written in java. The sections in which the Java code is written are called activity.

Java is;

* a robust language. Most programming errors can be understood at the writing stage.
* completely object-oriented. It is very easy to make, use, develop and transfer the object to another location.
* is a secure language. Virus cannot be made with Java language. No virus can infect a Java program. Every move a Java program makes is tracked. A malicious program cannot perform an operation if it is not allowed. This feature is very important in an environment where everyone can access someone else's machine, such as the Internet. This is why many companies with high security requirements prefer Java.
* an 'interpretive' language. This means that the commands of a Java program are translated into the format that the machine understands while it is running.
* dynamic. The units used (libraries, modules or classes) are connected to each other at run time.

## 2.2 Php

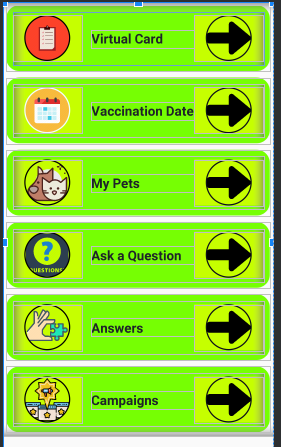
Web Service Codes Written in Php.

* Php a popular general-purpose scripting language that is especially suited to web development.
* PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of a HTTP response. Various web template systems, web content management systems, and web frameworks exist which can be employed to orchestrate or facilitate the generation of that response.
* PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications and robotic drone control.

# 3.PAGE DESIGNS AND FIGURES

## 3.1 Home Fragment

There are many functions to be used throughout the project in this section.

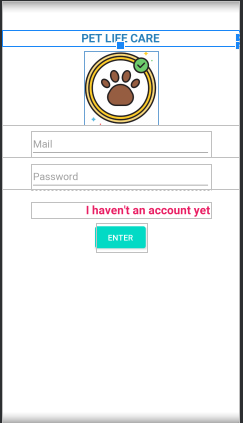


**Figure 1\_Home Fragment**

* Virtual Card is seen first. Here pet owners can view their pets and their vaccines.
* The second layout is Vaccination Date. In this section, pet owners can view the vaccine dates of their pet.
* My pets section is available for users to view only their pets.
* With Ask a question part, the user will be able to ask the veterinarian if there is any problem.
* The answers to the questions will appear in the answers section if the veterinarian answers. Pet owners will only be able to ask their questions in this area, which allows in-app messaging without disturbing their veterinarian's private number.
* Finally, there is the campaigns section. In this section, the veterinarian is informed about the products he sells. In this way, it will be possible for the veterinarian to earn more while pet owners shop for more affordable prices.

## 3.2 Entering Page

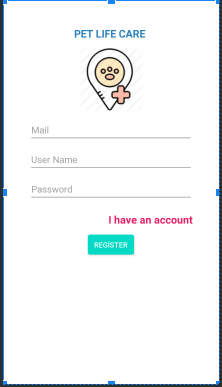
As with almost every application, a login screen is designed in this application. The registered user can login to the system by entering his / her email address and password with the page you have seen. If the user does not have a registered account, user can click on the I haven't account tab and the registration screen will be directed.



**Figure 2\_Entering Page**

## 3.3 Register Page

This page is coded for users who have not yet registered to the system. Here, information such as mail, user name, password is received from the user. Then, by clicking the register button, the user completes the registration process. If the user has accidentally clicked to this page even though he has a registered account, he clicks the I have an account tab and returns to the login page.



**Figure 3\_Register Page**

# 4.ANDROID STUDIO DETAILS

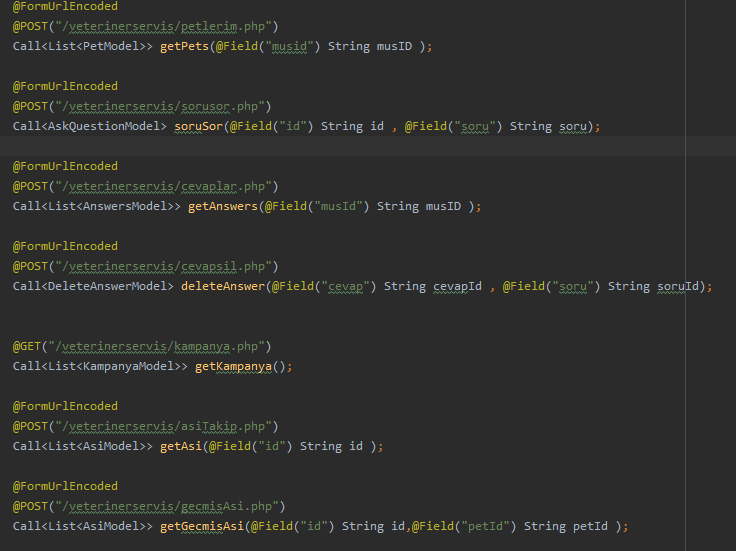
## 4.1 Retrofit

Retrofit is a REST Client for Java and Android. It makes it relatively easy to retrieve and upload JSON (or other structured data) via a REST based webservice. In Retrofit ,user configure which converter is used for the data serialization. Typically for JSON, users use GSon, but who can add custom converters to process XML or other protocols. Gson was also used in this project.. Retrofit uses the OkHttp library for HTTP requests.

The following three classes are mainly used to work with Retrofit:

* Model class used as JSON model
* Interface that identifies possible HTTP operations
* Retrofit.Builder class - Example that uses the interface and the Builder API to allow defining the URL endpoint for HTTP operations.

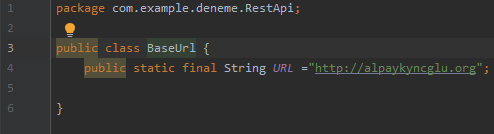
**4.1.1 RestApi**



**Figure 4\_RestApi**

In the above image, only a part of one of the classes written for the implementation of the retrofit library appears in the project. As can be seen, thanks to the post and get methods and the use of the model class, data flow is provided on the android and the web.

### 4.1.1 BaseUrl

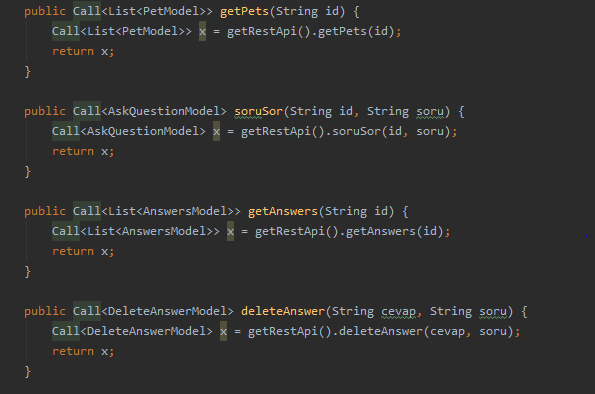


**Figure 5\_Base Url**

The purchased domain name was written in this class. While all web android interaction is provided, this class is called and the process is performed. The php extension expressions specified in the Restapi are written and encoded in the file manager on the control panel. This url must be called when accessing each of these specific php files.

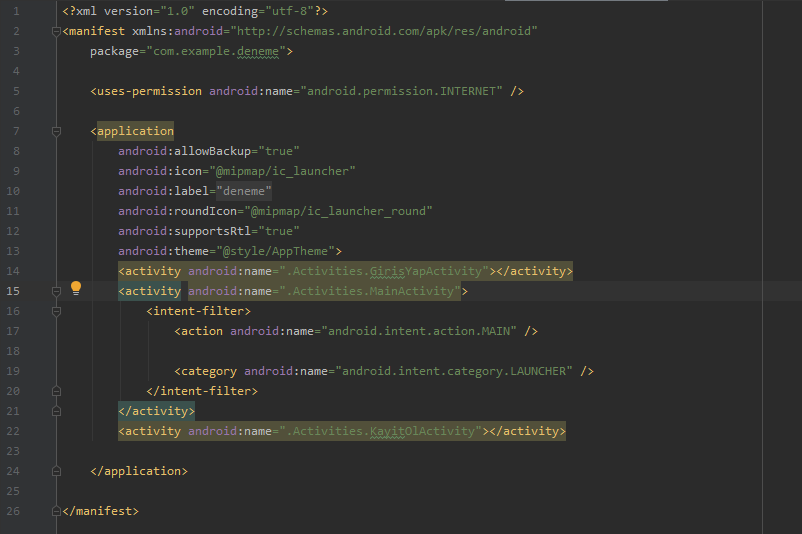
### 4.1.2 ManagerAll

The user will click on some buttons that will require data extraction from the database while performing operations on android. The Managerall class specifies what data this data is on the database. As it can be seen in the image, some expressions such as id, question, questionId are actually expressions determined when creating tables in the database

. 

**Figure 6\_Manager All**

## 4.2 Manifest,Permission and First Interface

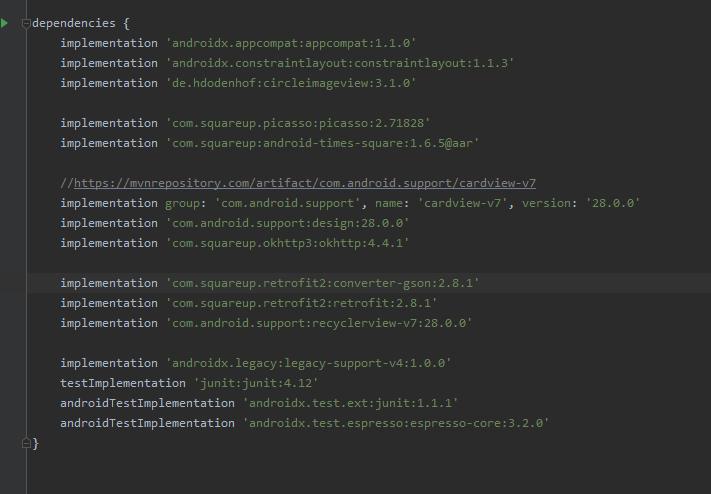


**Figure 7\_Manifest**

In this project, since the application has a web connection, the internet permission has been defined in the manifest section for the application to run smoothly.

Another important thing is which activity will run first when the program is opened. As can be seen, the first mentioned activity is the entrance activity. While writing the backend side in the program, the codings were made within a certain plan. Otherwise, there would be a disconnection in interfaces.

## 4.3 Libraries



**Figure 8\_Libraries**

This image shows all libraries used throughout the project;

Picasso library is used to import images from the database over the web.

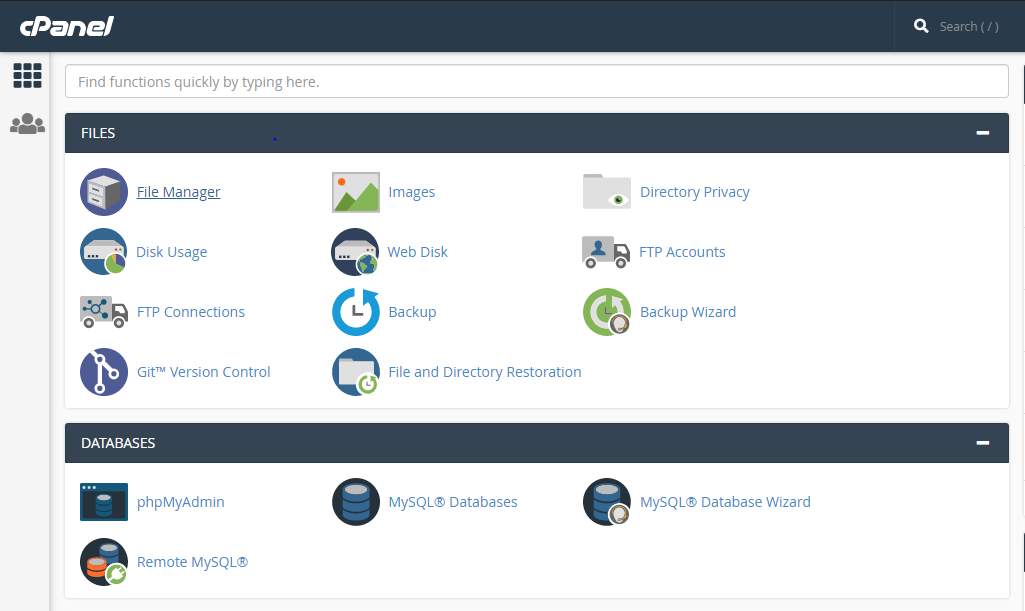
The circle imageview library was used to make the images used in homefragment look tidy and better in a certain template.

When some libraries are researched, they are first presented to the user as 'compile'. However, when library added as a compile, it will be seen that a line is formed on the compile text and it cannot be applied to the project. For this reason,expression of implemeantation should be used instead of compile statement.

Tabs such as pets and vaccines written in fragment are written in cardview. In this way, a better view is provided. In addition, many layouts were used in the XML side of the project. Thanks to the use of a different type of title from time to time also increased the readability of the project.

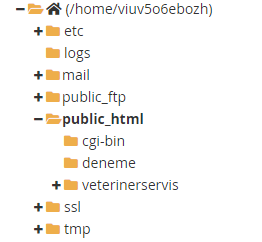
# 5.WEB SERVICES

This section is complementary to what has been described so far.



**Figure 9\_Control Panel**

This image shows the control panel interface inside the hosting. All the codes to be written for the project by clicking the File Manager section are defined in this section.



**Figure 10\_File Manager**

The codes required for the project are written under the public html header. If the codes are not collected under the correct title, the project will not work.

## 5.1 Web Service Content

### 5.1.1 Settings



**Figure 11\_Settings**

The codes providing the necessary integration for the web service are above. This file has been called in many code pages that follow. The reason for this is that the code on this page provides access to the control panel and database.

In order to provide this connection, four parameters must be sent to the system and confirmed. These are user name, password, servername and database name. These parameters are sent with mysqli\_ connect. There is a situation that may vary depending on host services. While doing this project, default value given for control panel is defined as servername. Thus, successful operation of the code was ensured.

### 5.1.2 Register Service

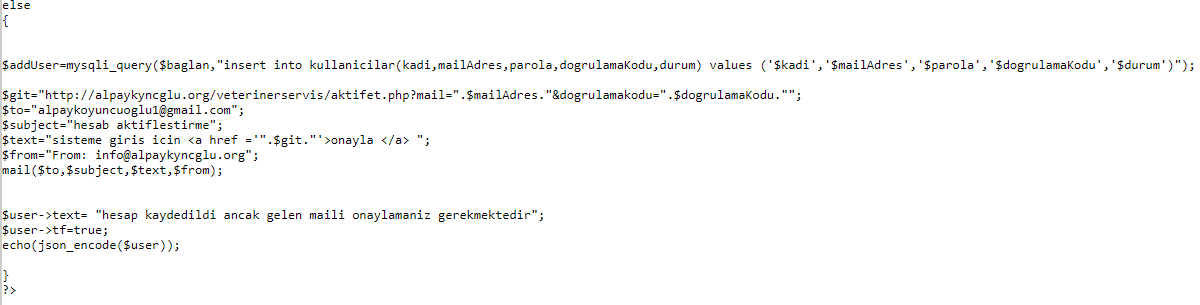


**Figure 12\_Register Service**

On the Android side, after the user enters the necessary information in the registration interface and clicks the registration button, the system should transfer this data to the database. Post methods shown here transmit the data transferred to the android interface to the database.

Control codes have been written to prevent my system to be registered with the same e-mail address or the same user name.

If the user enters their already existing name or e-mail address information, a warning message is returned as seen above.



**Figure 13\_Register Service 2**

If the same name or mail address is not in the system, data is added to the database. This is done with insert into command.

The system sends confirmation mail immediately after adding to the database.This is achieved by the mail function seen above.

With 'to', the e-mail address to be sent is specified.

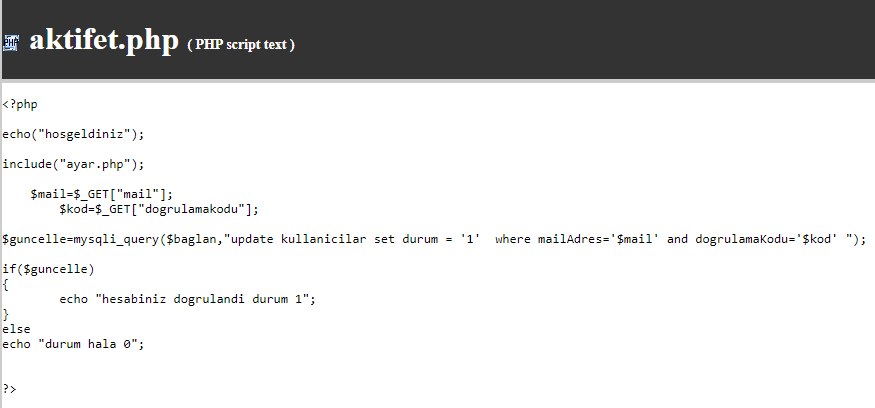
A mail adress for testing purposes has been entered here. In normal condition, the piece of code that will pull the mail address data entered by the user from the database is written here.

Subject section specifies the mail header.

Who will send the mail is written 'from' part.

If the host received contains mail service, the host mail is received by going to the required section on the control panel.

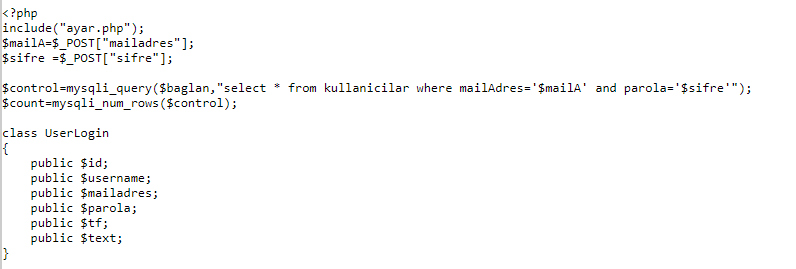
### 5.1.2 Activate



**Figure 14\_Activate Service**

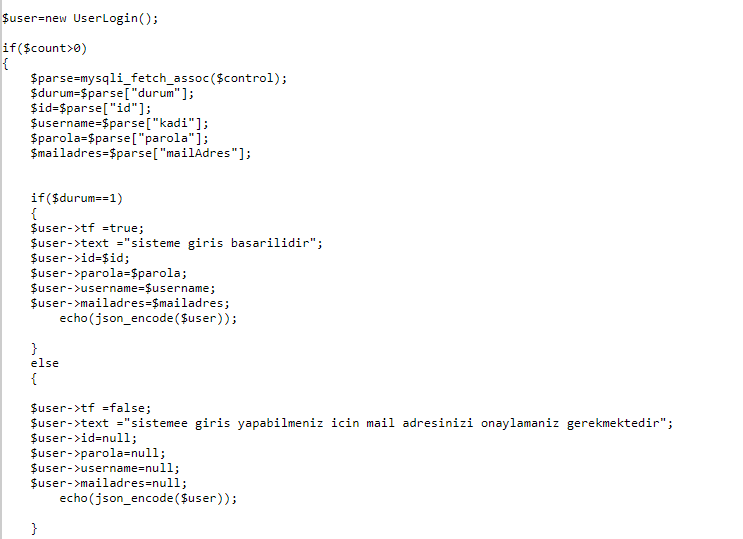
If the user clicks on the confirmation e-mail sent to the e-mail address and the account is confirmed. For this to happen, get method is used. In this way, the mail and user name information to be activated is received and the status information is converted from zero to one.

### 5.1.4 Login



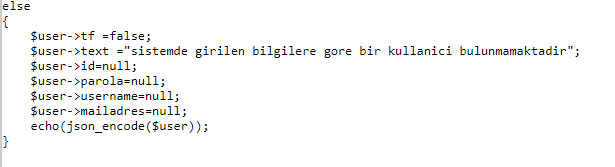
**Figure 15\_Login Service**

When the user enters their e-mail and password information, they are controlled by the system.



**Figure 16\_Login Service 2**

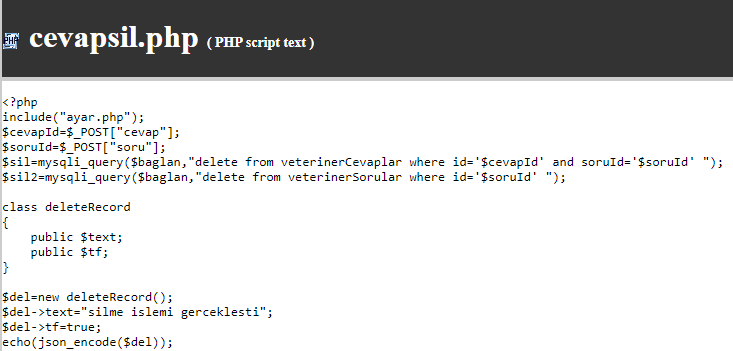
If there is a record in the entered values, count value is 1 and input is provided in the first if. If the user has activated the confirmation mail sent to his / her email address, the second if is entered and the successful login message is displayed on the screen. If the confirmation mail is still not active, go to the else section and system inform the user that the account has not yet been approved.



**Figure 17\_Login Service 3**

If there is no user of the information entered by the user, the above codes will work and you will be warned that there is no such user.

### 5.1.5 Delete Answer



**Figure 18\_Delete Service**

Here is how to delete a data from the database. Connecting with settings.php, the use of the post method has already been described. A different query is seen here. After the 'delete from' statement, the name of the table to be deleted is written. Then, which id's questions will be deleted is specified in the ongoing section.

# RESOURCES

[https://www.vogella.com/tutorials/](https://www.vogella.com/tutorials/Retrofit/article.html#:~:text=Retrofit%20is%20a%20REST%20Client,via%20a%20REST%20based%20webservice.&text=Typically%20for%20JSON%20you%20use,OkHttp%20library%20for%20HTTP%20requests.)

<https://abhiandroid.com/ui/adapter>

<https://developer.android.com/guide/components/fragments>

<https://www.quora.com/What-is-Gson-and-how-does-it-work-in-Android>

<https://searchapparchitecture.techtarget.com/definition/RESTful-API>

<https://denizirgin.com/rest-ve-restful-web-servis-kavram%C4%B1-30bc4400b9e0>

<http://okanyuksel.site/retrofit-nedir-nasil-kullanilir/>

<https://medium.com/gdgtekirdag/android-ile-retrofit-kullan%C4%B1m%C4%B1-spacexdata-71a5117666d1>

<https://www.iconfinder.com/>

<https://medium.com/cool-digital-solutions/android-3rd-party-library-kullan%C4%B1m%C4%B1-cac30e1a4f68>

<https://www.color-hex.com/color-palettes/>