Introduction

It is a simple multi-platform application, that helps users find a suitable laptop by mentioning the desired budget using voice search. The user has to give a budget within which they want the laptop. The user will be given three suitable options for laptops that best match their budget. The user can then click on a link on the laptop they like the most among three, which will take the user to the page on an online website, where he/she can buy the chosen laptop.

Contributors

Name	Email
Farishta Jayas Kinjol	farishtajayas@gmail.com
Muhammad Irfanul Haque	haque.irfanul.m@gmail.com
Syed Maaher Hossain	maaher1145@gmail.com
Abidur Rahman	abutat123@gmail.com

Project Status

List of proposed Use Cases	List of completed use cases
Use voice command to search a laptop	Use voice command to search a laptop
Go with the primary selection or opt for secondary options	Go with the primary selection or opt for secondary options
Buy the laptop directly from the link	Buy the laptop directly from the link
Search for laptop using seamless conversational speech	Search for laptop using specific voice commands
User can save laptops to see later	Not implemented

Overall Final System design

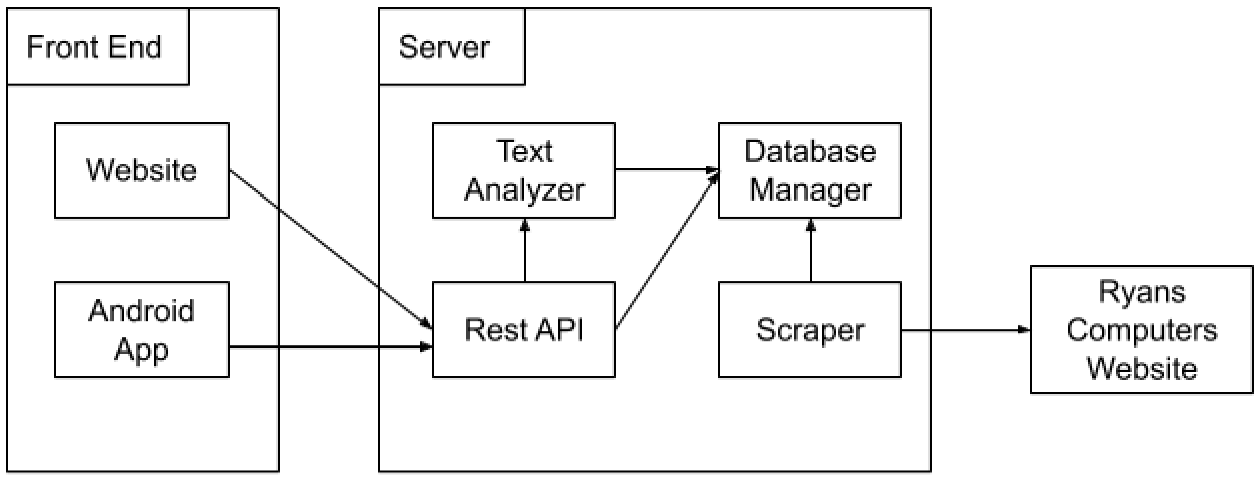


Fig. Overall Final System Design

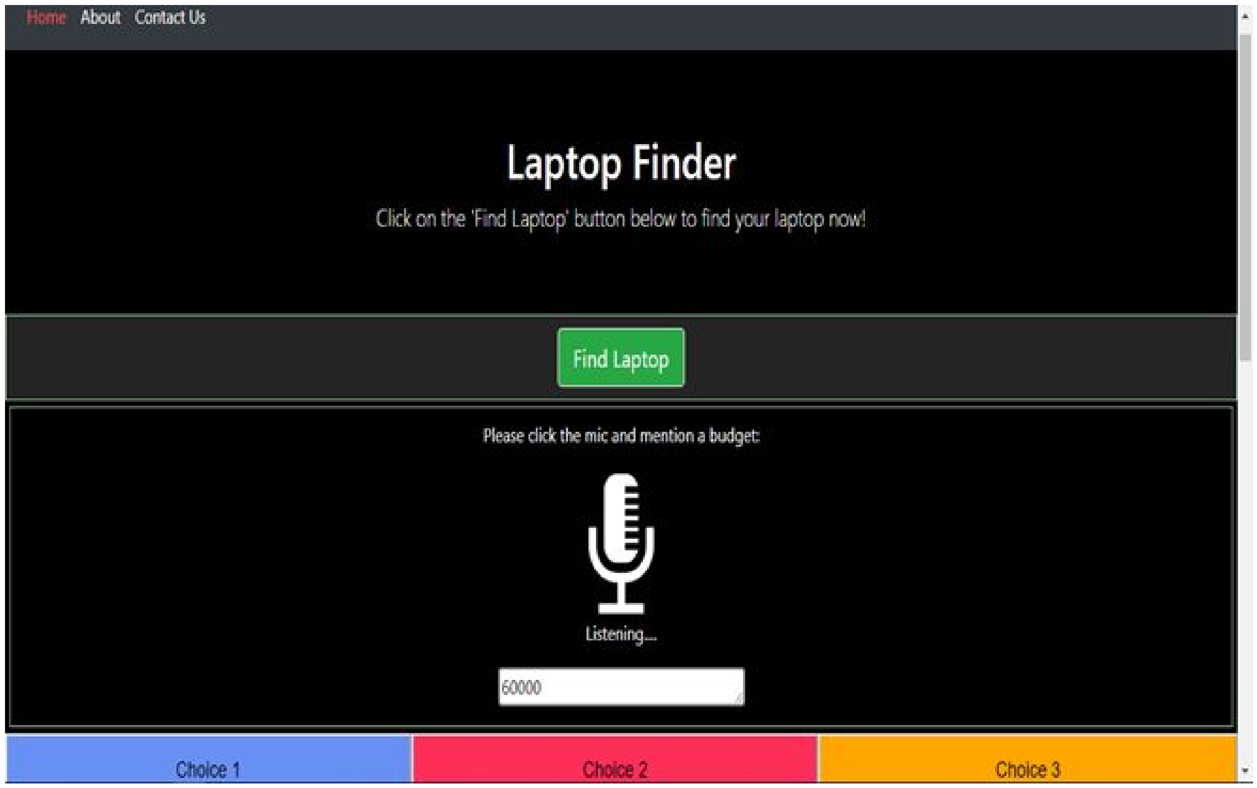
Detail of different platforms

Platform 1: Web Application

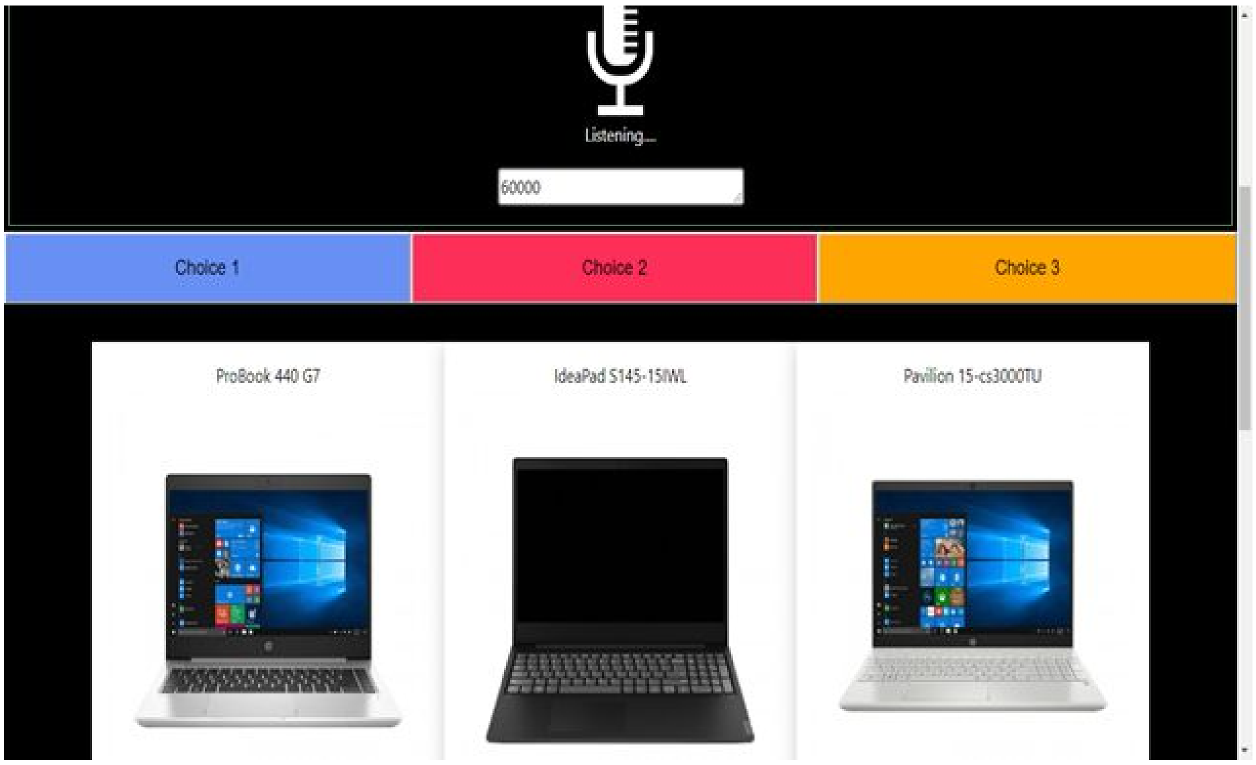
The web application is one of the frontends which takes voice input from the user and communicates with the server to find suitable laptops and show them to the user.

Use voice command to search a laptop:

The user has to tap on the Find Laptop button and then the mic and mention a budget to search for the desired laptops

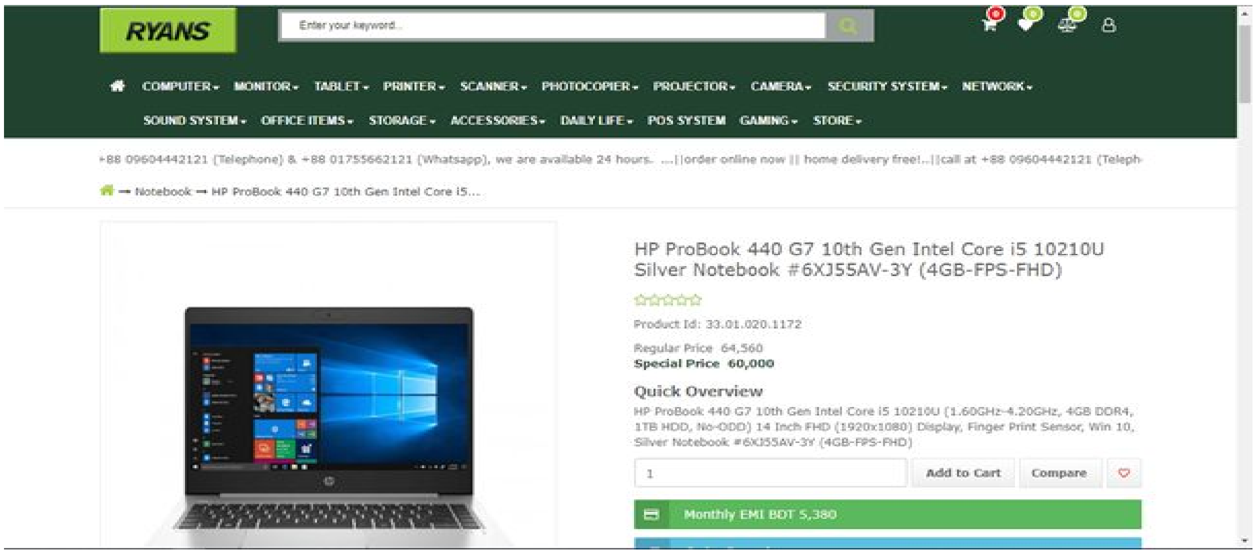

Home
About
Contact Us
Laptop Finder
Click on the 'Find Laptop' button below to find your laptop now!
Find Laptop
Please click the mic and mention a budget:
Listening....
60000
Choice 1
Choice 2
Choice 3

Go with the primary selection or opt for secondary options:



The user is given a total of three choices so that if the user is not satisfied with the first option, he/she can go for the two other alternatives

Buy the laptop directly from the link:


RYANS
Enter your keyword...
COMPUTER MONITOR TABLET PRINTER SCANNER PHOTOCOPIER PROJECTOR CAMERA SECURITY SYSTEM NETWORK
SOUND SYSTEM OFFICE ITEMS STORAGE ACCESSORIES DAILY LIFE POS SYSTEM GAMING STORE
Add to Cart
Compare

+88 09604442121 (Telephone) & +88 01755662121 (Whatsapp), we are available 24 hours. ...||order online now || home delivery free!..||call at +88 09604442121 (Teleph

→ Notebook → HP ProBook 440 G7 10th Gen Intel Core i5...

HP ProBook 440 G7 10th Gen Intel Core i5 10210U Silver Notebook #6XJ55AV-3Y (4GB-FPS-FHD)

Product Id: 33.01.020.1172

Regular Price 64,560
Special Price 60,000

Quick Overview
HP ProBook 440 G7 10th Gen Intel Core i5 10210U (1.60GHz-4.20GHz, 4GB DDR4, 1TB HDD, No-ODD) 14 Inch FHD (1920x1080) Display, Finger Print Sensor, Win 10, Silver Notebook #6XJ55AV-3Y (4GB-FPS-FHD)

1

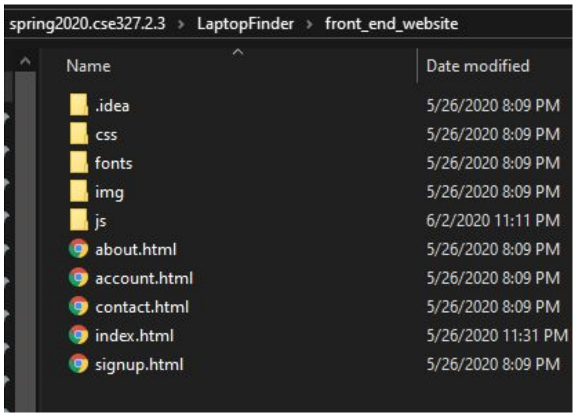
Monthly EMI BDT 5,380

The user has to click on the image of the desired laptop on the website, which will take the user to the online page of Ryan’s, where he/she can buy the laptop.

Website Codes

The website codes can found in the following folder

LaptopFinder\front_end_website



spring2020.cse327.2.3 › LaptopFinder › front_end_website

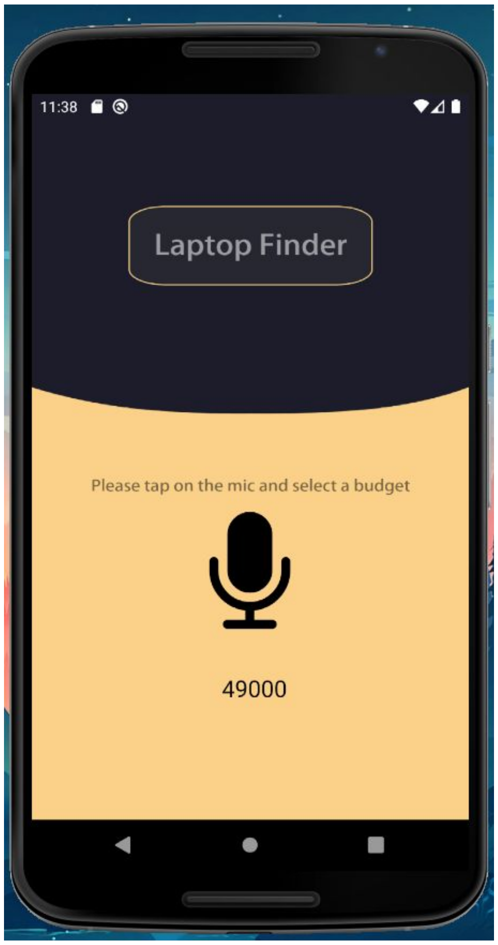
Name	Date modified
.idea	5/26/2020 8:09 PM
css	5/26/2020 8:09 PM
fonts	5/26/2020 8:09 PM
img	5/26/2020 8:09 PM
js	6/2/2020 11:11 PM
about.html	5/26/2020 8:09 PM
account.html	5/26/2020 8:09 PM
contact.html	5/26/2020 8:09 PM
index.html	5/26/2020 11:31 PM
signup.html	5/26/2020 8:09 PM

Platform 2: Android Mobile Application

The android version of Laptop Finder is essentially as powerful as the web version, but it can be run on a mobile Android device. Running it is fairly simple. When the Android device is opened the user will be greeted with the following home screen:

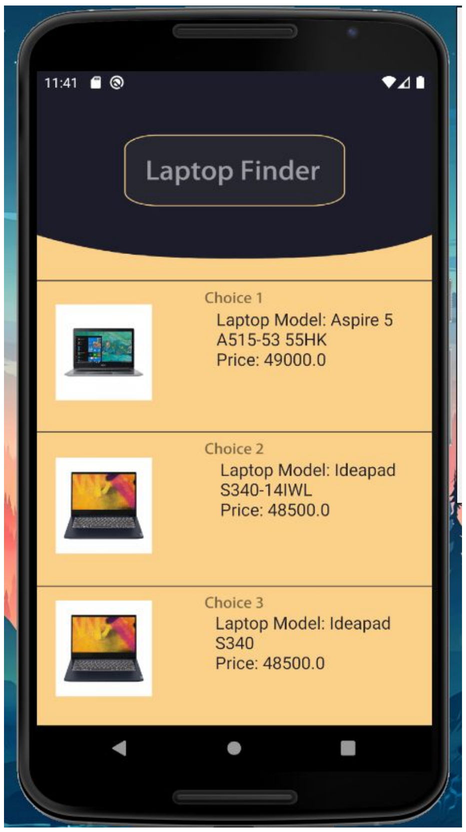
Use voice command to search a laptop:

All the user has to do is tap on the mic and input the budget with his vocal cords. (Well, by saying their budget in the form of numbers out loud).



Go with the primary selection or opt for secondary options

_After inputting the budget, the user will be met with our Results page where three laptop options will be displayed along with the laptop model and the price.



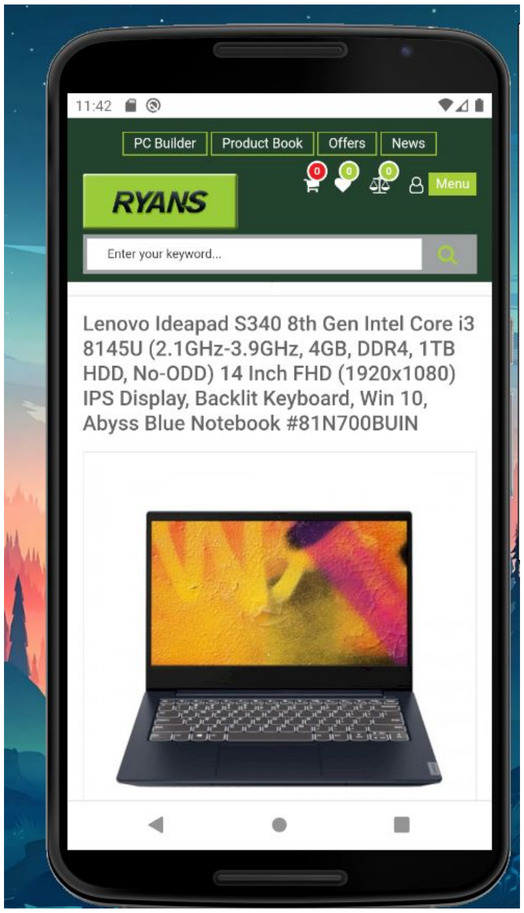
Choice 1
Laptop Model: Aspire 5 A515-53 55HK
Price: 49000.0

Choice 2
Laptop Model: Ideapad S340-14IWL
Price: 48500.0

Choice 3
Laptop Model: Ideapad S340
Price: 48500.0

Buy the laptop directly from the link:

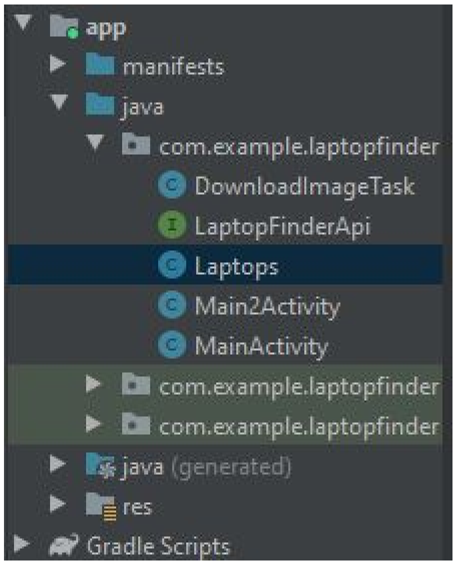
Each laptop image is a link to an external website, where the user can buy the laptops online.



Lenovo Ideapad S340 8th Gen Intel Core i3 8145U (2.1GHz-3.9GHz, 4GB, DDR4, 1TB HDD, No-ODD) 14 Inch FHD (1920x1080) IPS Display, Backlit Keyboard, Win 10, Abyss Blue Notebook #81N700BUIN

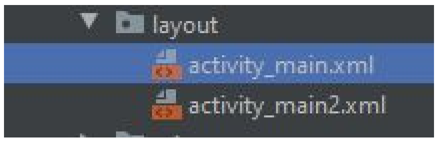
-

Android Code Details:



```
▼ app
    ▶ manifests
    ▼ java
        ▼ com.example.laptopfinder
              DownloadImageTask
              LaptopFinderApi
              Laptops
              Main2Activity
              MainActivity
        ▶ com.example.laptopfinder
        ▶ com.example.laptopfinder
    ▶ java (generated)
    ▶ res
▶ Gradle Scripts
```

All the activity codes can be found under the Java/com.example.laptopfinder.



```
▼ layout
      activity_main.xml
      activity_main2.xml
```

The layout codes can be found under

1) res/layout/actvitvity_main.xml
2) res/layout/actvitvity_main2.xml

API Details

Django Server:

LaptopFinder/Server/

The server is hosted on the local network. For the website, it is assumed that the server and website are running on the same machine, so we run the server in the localhost.

Running the server for the website:

```
cd "LaptopFinder/Server"
python manage.py runserver
```

Running the app on either the virtual machine within the android studio or a standalone android device requires the server to be accessible to other devices on the network. So, it needs to be hosted on the machine's local IP (192.168.x.x) instead of the localhost (127.0.0.1).

Running the server for the app:

```
cd "LaptopFinder/Server"
python manage.py runserver YOURMACHINEIP:8000
```

Rest API

The frontend and backend of this application are connected using a Rest API. The following table shows the endpoints and what each of them return.

API Endpoints:

	Endpoint	**Response**
GET	api/laptops_by_brand/{laptop_brand}	Get list of laptops filtered by brand
GET	api/laptops_by_analyzer/{text}	Get list of laptops through analyzed text (currently only filters by brand)
GET	api/laptops_by_price/{laptop_price}	Get list of all laptops at or below laptop_price (sorted in descending order of price)

Scraper

LaptopFinder/Server/apps/laptop_scraper/

The scraper populates our database with laptop information from Ryans Computers. Upon running, it fetches data from each laptop on the website, creates “djangoitem” objects, which correspond with the Django models in the database, and saves them in the database.

Running the scraper:

```
cd "LaptopFinder/Server/apps/laptop_scraper"
scrapy crawl ryans_scraper
```

Database

The database has been designed with three tables. User, Laptop Properties, Record. The figures below describe the purpose and structure of these tables.

Table	**Purpose**
User	To contain user data
Laptop Properties	To contain information regarding laptops
Record	To contain records of which laptops a user has saved for future reference

Fig. Database - Table Purpose

Field	**Description**
user_id	Unique user id
first_name	User first name
last_name	User last name

Fig. Database - User Table

Field	**Description**
lapID	Unique laptop id
lapBrand	Laptop brand
lapModel	Laptop model
ram	Amount and type of RAM
processor	Processor make and model of laptop
gpu	GPU make and model of laptop
purpose	Purpose (Gaming/Workstation/etc)
price	Price of laptop in BDT
url	Link to where to buy the laptop from
image_url	Link to image on the seller's website
rating	Rating of laptop

Fig. Database - Laptop Property Table

Field	**Description**
recordID	Unique record id
u_id	User id (foreign key from User table)
l_id	Laptop id (foreign key from Laptop Property Table

Fig. Database - Record Table

Database Manager

LaptopFinder/Server/apps/database_manager/

The database manager is a Django app that contains all the database related code. The database described above has been implemented using the Django default ORM. The relevant code for the Django models can be found in “**models.py**”. The database manager also contains APIView classes which enable administrator access to read and modify the database and handles the queries that the Text Analyzer app needs to get laptop information from the database. The implementation of this can be found in “**api.py**”

Text Analyzer

LaptopFinder/Server/apps/text_analyzer/

The text analyzer is a Django App with two APIView classes that return rest API responses containing laptop information from the database. The classes are Text_Analyzer and Laptops_By_Price. The relevant code for the classes can be found in “**api.py**”.