الجامعه المصريه للتعلم االلكتروني الهليه

Faculty of Computer & Information Technology.

Medical community

By:

Abanoub Eshak Azmy	20-01036
Marina Kamel Abd eltwab	20-00931
Manar Mohammed Awny	20-01117
Beshoy Mokhlis Agaiby	20-01086
Abanoub Gaber Zikry	20-00517
Abanoub Emad Felix	20-01523
Andrew Samer lahzy	20-00437

Under Supervision of:

Dr. Walaa EL Sayed

Professor of Computer and Information Technology Egyptian E-Learning University

Eng. Salma Montaser

Teaching Assistant in Faculty of Computer and Information Technology Egyptian E-Learning University

Assiut 2024

Acknowledgment

In the name of Allah, the Most Gracious and the Most Merciful Alhamdulillah, all praise Allah for the strengths and His blessing in completing this Project. In performing our project, we had to take the help and guidelines of some respected persons, who deserve our greatest gratitude. The completion of this project gives us much Pleasure. We would like to present our supervisors:

Dr. Walaa EL Sayed

Computer and Information Technology Program Director – EELU

Eng. Salma Montaser

Faculty of Computers and Information Technology- EELU

For giving us a good guideline for the Project throughout numerous consultations. We would also like to extend our deepest gratitude to all those who have directly and indirectly guided us in doing this Project.

In addition, a thank you to thank Eng. Salma Montaser introduced us to the methodology of work, and whose passion for the "underlying structures" had lasting effects. We also thank our university EELU for consent to include copyrighted pictures as a part of our graduation project.

Many people, especially our classmates and team members, have made valuable comments and suggestions on this proposal which inspired us to improve our project. We thank all the people for their help directly and indirectly in completing our project.

Abstract

Explain application :

The idea of the project is the initial diagnosis of the patient and it includes the specialty of internal medicine, as we have compiled a data set from doctors and it includes 12 symptoms, age, gender, and a description of the disease and most internal diseases, where the patient registers on the application and enters the username, email, gender, age, and password, and confirms the password if any. He enters for the first time, or sign in if he has previously logged in before, where he enters the email and password, and this information is recorded on the profile page, then he is transferred to the home page, then he clicks on the ask me page and writes all the symptoms he feels, where the chat responds. The disease he is afflicted with, his description of the disease, his age, and gender.

Summary for Internal diseases :

Internal diseases encompass a wide range of conditions affecting various organs and systems within the body. Understanding the pathophysiology, etiology, symptoms, diagnosis, treatment, and prognosis of these diseases is crucial for effective management and improving patient outcomes. The complexity of these diseases often requires a multidisciplinary approach to diagnosis and treatment, involving various specialists and healthcare providers.

Internal diseases : refer to a broad category of ailments that primarily affect the internal organs and systems of the body, such as the heart, lungs, liver, kidneys, and gastrointestinal tract. These diseases often involve complex interactions between various biological systems and can manifest through a wide range of symptoms.

Table of content

Chapter 1

Introduction

Introduction:

Internal diseases, also known as internal medicine or internal organ diseases, encompass a wide range of medical conditions that affect the internal organs and systems of the body. Unlike external injuries or visible illnesses, these diseases manifest within the body's intricate network of organs, tissues, and physiological processes. Understanding internal diseases is fundamental to modern medicine as they often pose complex diagnostic challenges and require specialized medical expertise for effective management.

The field of internal medicine is vast and covers diverse areas such as cardiology (heart diseases), gastroenterology (digestive system disorders), pulmonology (respiratory diseases), nephrology (kidney diseases), endocrinology (hormonal disorders), and more. Internal diseases can result from various factors including genetics, environmental influences, lifestyle choices, infections, immune system dysfunctions, and aging processes.

Diagnosing internal diseases typically involves a comprehensive approach that includes detailed medical history, physical examinations, laboratory tests, imaging studies (like X-rays, CT scans, and MRI), and sometimes, invasive procedures such as biopsies or endoscopic examinations. The goal is to identify the underlying cause of symptoms, determine the extent of organ involvement, and formulate an appropriate treatment plan.

Treatment of internal diseases varies widely depending on the specific condition, its severity, and the patient's overall health. Treatment modalities may include medications (such as antibiotics, anti-inflammatory drugs, immunosuppressants), lifestyle modifications (dietary changes, exercise regimens, smoking cessation), surgical interventions (like organ transplants, tumor removal), and supportive therapies (such as physical therapy, counseling, palliative care).

Preventive healthcare plays a crucial role in managing internal diseases by focusing on risk factor modification, early detection of conditions through screenings and check-ups, vaccination programs, and patient education about healthy living practices. Advances in medical research, technology, and pharmaceuticals continue to drive innovation in the diagnosis, treatment, and prevention of internal diseases, improving patient outcomes and quality of life.

In summary, internal diseases represent a complex and diverse spectrum of medical conditions that require comprehensive medical care, interdisciplinary collaboration among healthcare professionals, ongoing research, and a patient-centered approach to promote health and well-being.

These diseases can vary widely in terms of causes, symptoms, treatments, and prognoses. Here are some abstract points to consider:

1. Complexity of Diagnosis: Internal diseases often present diagnostic challenges because symptoms may not be immediately visible or may overlap with other conditions. This complexity requires thorough medical history-taking, physical examinations, and often, specialized tests like imaging or laboratory investigations.

2. Multifactorial Causes: Many internal diseases have multifactorial causes, meaning they result from a combination of genetic, environmental, lifestyle, and sometimes unknown factors. Understanding these underlying causes is crucial for developing effective treatment and prevention strategies.

3. Systemic Impact: Internal diseases can have systemic effects, meaning they can affect multiple organs or systems within the body. For example, conditions like diabetes mellitus or hypertension not only impact specific organs like the pancreas or heart but also have widespread effects on overall health.

4. Treatment Challenges: Treatment approaches for internal diseases can vary widely depending on the specific condition and its severity. Treatment may involve medications, lifestyle modifications, surgical interventions, or a combination of these. Managing chronic internal diseases often requires long-term care and monitoring.

5. **Preventive Focus:** Given the complexity and impact of many internal diseases, there is a growing emphasis on preventive healthcare measures. This includes promoting healthy lifestyles, early detection of risk factors, regular screenings, and vaccination programs to prevent certain diseases.

6. **Research and Innovation:** Advances in medical research and technology play a vital role in understanding and managing internal diseases. From genetic studies to cutting-edge diagnostic tools and therapies, ongoing research continues to expand our knowledge and improve patient outcomes.

7. **Psychosocial Impact:** Internal diseases can also have significant psychosocial impacts on patients and their families. Chronic conditions, in particular, may require ongoing emotional support, education about the condition, and access to community resources for holistic care.

Understanding the abstract concepts surrounding internal diseases helps us appreciate the complexity of medical science and the interdisciplinary nature of healthcare. It also underscores the importance of holistic approaches to patient care, considering not just the physical aspects but also the psychological, social, and environmental factors influencing health and wellness.

Internal medicine is a branch of medicine that focuses on the prevention, diagnosis, and treatment of a wide range of diseases in adults. The exceptional clinicians, investigators, educators, trainees, and staff in the Department of Internal Medicine at Yale form an internationally renowned department in one of the world’s top medical schools.

Embedded in a remarkable basic science environment at Yale, our more than 3,300 members work in a collaborative culture with numerous opportunities for breakthrough interdisciplinary and translational research.

Internal medicine is a discipline that applies to all clinical branches. Most of the problems of patients who present at health establishments are related to the internal diseases department. It covers many fields including Upper and lower respiratory tract diseases, hypertension, stomach-intestinal system, kidney diseases, thyroid, diabetes, blood, oncology-cancer rheumatism diseases, and many more. Additionally, the internal diseases department also covers outpatients, inpatients, emergency services, and check-up services. The Near East University Hospital Internal Diseases Department, in addition to conducting check-ups and the necessary tests, performs treatments for inpatients, conducts consultations, and does pre-surgery preparations and necessary interventions.

Check-up and Preventative Medicine: as a preventative medicine service, our specialist doctors determine the risks that we may face, then give recommendations in terms of which precautions should be taken The preventative medicine services available at our department include Risk group HIV test, sexually transmitted diseases tests, blood pressure control, cholesterol control, diabetes test, thyroid function test, osteoporosis test, determination of obesity causes, and hepatitis test. The check-up services, which are an important part of preventative medicine services, are provided with personalized programs depending on the patient's age, gender, and disease risks.

Chapter 2

Background

2.1 Mobile Application (Flutter & Firebase)

A mobile application, also referred to as a mobile app or simply an app, is a computer program or software application designed to run on a mobile device such as a phone, tablet, or watch. Apps were originally intended for productivity assistance such as email, calendar, and contact databases, but the public demand for apps caused rapid expansion into other areas such as mobile games, factory automation, GPS and location-based services, order-tracking, and ticket purchases so that there are now millions of apps available. Apps are generally downloaded from application distribution platforms that are operated by the owner of the mobile operating system, such as the App Store (iOS) or Google Play Store. Some apps are free, and others have a price, with the profit being split between the application's creator and the distribution platform. Mobile applications often stand in contrast to desktop applications which are designed to run on desktop computers, and applications which run in mobile browsers rather than directly on the mobile device.

2.1.1 Flutter.

Flutter is a cross-platform UI toolkit that is designed to allow code reuse across operating systems such as iOS and Android, while also allowing applications to interface directly with underlying platform services. The goal is to enable developers to deliver high-performance apps that feel natural on different platforms, embracing differences where they exist while sharing as much code as possible. In this project, they develop a mobile application using the Flutter platform to detect images by training the DL model on this application and recognize it contains any letter of the alphabet, words, words, or sentences easily.

Advantages of Flutter.

Flutter comes with beautiful and customizable widgets for high-performance and outstanding mobile applications. It fulfills all the custom needs and requirements. Besides these, Flutter offers many more advantages as mentioned below Dart has a large repository of software packages that lets you extend the capabilities of your application. Developers need to write just a single code base for both applications (both Android and iOS platforms). Flutter may be extended to another platform as they'll in the future.

Flutter needs less testing. Because of its single code base, it is sufficient if they write automated tests once for both platforms.

Flutter runs the code on Android:-

The engine's C and C++ code are compiled with Android's NDK. The Dart code (both the SDKs and yours) is ahead-of-time (AOT) compiled into native, ARM, and x86 libraries. Those libraries are included in a "runner" Android project, and the whole thing is built into a .apk. When launched, the app loads the Flutter library. Any rendering, input, event handling, and so on, is delegated to the compiled Flutter and app code. This is similar to the way many game engines work.

During debug mode, Flutter uses a virtual machine (VM) to run its code to enable stateful hot reload, a feature that lets you make changes to your running code without recompilation. You’ll see a “debug” banner in the top right-hand corner of your app when running in this mode, to remind you that performance is not characteristic of the finished release app.

Flutter runs the code on IOS:-

The engine’s C and C++ code are compiled with LLVM. The Dart code (both the SDK’s and yours) are ahead-of-time (AOT) compiled into a native, ARM library. That library is included in a “runner” iOS project, and the whole thing is built into an. ipa. When launched, the app loads the Flutter library. Any rendering, input event handling, and so on, are delegated to the compiled Flutter and app code. This is similar to the way many game engines work. During debug mode, Flutter uses a virtual machine (VM) to run its code to enable stateful hot reload, a feature that lets you make changes to your running code without recompilation. You’ll see a “debug” banner in the top right-hand corner of your app when running in this mode, to remind you that performance is not characteristic of the finished release app.

Dart Flutter.

Dart is an open-source general-purpose programming language. It was originally developed by Google. Dart is an object-oriented language with C-style syntax. It supports programming concepts like interfaces and classes, unlike other programming languages Dart doesn’t support arrays. Dart collections can be used to replicate data structures such as arrays, generics, and optional typing.

The variable is named storage location and Data types simply refer to the type and size of data associated with variables and functions.

Dart language supports the following data types:-

➢ **Numbers**: - It is used to represent numeric literals Integer and Double.

➢ **Strings**: - It represents a sequence of characters. String values are specified in either single or double quotes.

➢ **Booleans**: - Dart uses the bool keyword to represent Boolean values true and false.

➢ **Lists and Maps**: - It is used to represent a collection of objects. A simple List can be defined as below.

Dart libraries: -

Many Dart libraries like.

➢ Built-in types, collections, and other core functionality for every Dart program (dart: core).

➢ Richer collection types such as queues, linked lists, hash maps, and binary trees (Dart: collection).

➢ Encoders and decoders for converting between different data representations, including JSON and UTF-8 (Dart: convert).

➢ Mathematical constants and functions, and random number generation (Dart: math).

➢ File, socket, HTTP, and other I/O support for non-they applications (dart: io).

➢ async and await are keywords that provide a way to make asynchronous operations appear synchronous (Dart: async await)

➢ Lists that efficiently handle fixed-sized data (for example, unsigned 8-byte integers) and SIMD numeric types (dart: typed_data).

2.1.2 Firebase.

Firebase is a Backend-as-a-Service (BaaS) app development platform that provides hosted backend services such as cloud storage, crash reporting, machine learning, remote configuration, and hosting for your static files. Firebase supports Flutter. It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google's infrastructure.

Key Features: -

➢ Authentication

➢ Realtime database

➢ Hosting

➢ Notifications

The Firebase Realtime Database lets you build rich, collaborative applications by allowing secure access to the database directly from client-side code. Data is persisted locally, and even while offline, real-time events continue to fire, giving the end user a responsive experience. When the device regains connection, the Realtime database synchronizes the local data changes with the remote updates that occurred while the client was offline, merging any conflicts automatically.

libraries: -

➢ Firebase Core.

➢ Cloud Fire Store.

2.2 Machine Learning

Machine learning is a branch of artificial intelligence (AI) and computer science that focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy. Machine Learning tutorial provides basic and advanced concepts of machine learning. Our machine-learning tutorial is designed for students and working professionals. This machine learning tutorial gives you an introduction to machine learning along with a wide range of machine learning techniques such as Supervised and Unsupervised learning. You will learn about regression, classification models, and clustering.

Classes of Machine Learning

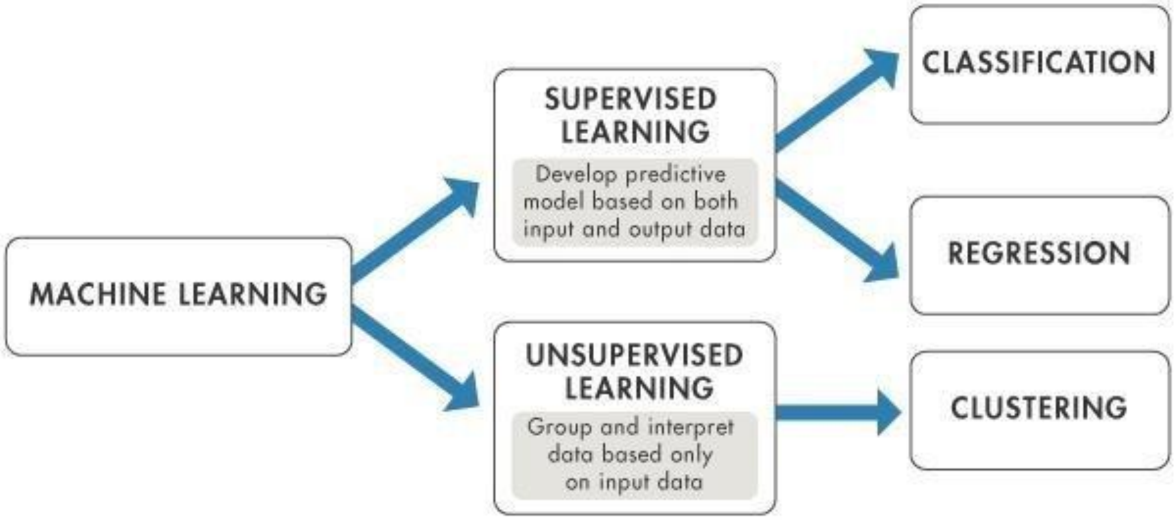

MACHINE LEARNING
SUPERVISED LEARNING
Develop predictive model based on both input and output data
UNSUPERVISED LEARNING
Group and interpret data based only on input data
CLASSIFICATION
REGRESSION
CLUSTERING

Figure Classification of Machine Learning

2.2.1

Supervised Learning.

Supervised learning addresses the task of predicting target 8 given input data. The targets, also commonly called labels, are generally denoted y. The input data points, also commonly called examples or instances, are typically denoted x. The goal is to produce a model fθ that maps an input x to a prediction fθ(x). In probabilistic terms, they typically are interested in estimating the conditional probability P(y|x). While it's just one among several approaches to machine learning, supervised learning accounts for the majority of machine learning in practice. Partly, that's because many important tasks can be described crisply as estimating the probability of some unknown given some available evidence:

2.2.1 Predict cancer vs not cancer, given a CT image.

2.2.2 Predict the correct translation in French, given a sentence in English.

2.2.3 Predict the price of a stock next month based on this month's financial reporting data.

Even with the simple description 'predict targets from inputs' supervised learning can take a great many forms and require a great many modeling decisions, depending on the type, size, and the number of inputs and outputs. For example, they use different models to process sequences (like strings of text or time series data) and for processing fixed-length vector representations.

Classification:

Classification is defined as the process of recognition, understanding, and grouping of objects and ideas into preset categories a.k.a "subpopulations." With the help of these pre-categorized training datasets, classification in machine learning programs leverages a wide range of algorithms to classify future datasets into respective and relevant categories. Classification algorithms used in machine learning utilize input training data to predict the likelihood or probability that the data that follows will fall into one of the predetermined categories. One of the most common applications of classification is for filtering emails into “spam” or “non-spam”, as used by today’s top email service providers.

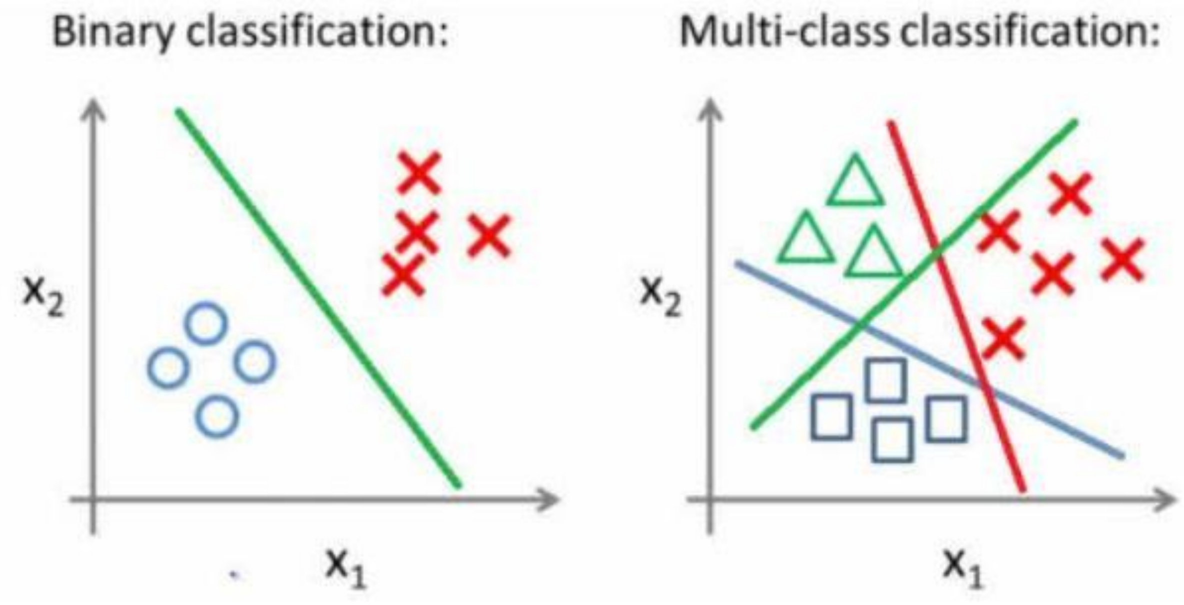


figure 2 Types of classification

In short, a classification is a form of “pattern recognition,”. Here, classification algorithms applied to the training data find the same pattern (similar number sequences, words or sentiments, and the like) in future data sets. They will explore classification algorithms in detail and discover how text analysis software can perform actions like sentiment analysis used for categorizing unstructured text by opinion polarity (positive, negative, neutral, and the like).

Regression: Perhaps the simplest supervised learning task to wrap your head around is Regression. Consider, for example, a set of data harvested from a database of home sales. They might construct a table, where each row corresponds to a different house, and each column corresponds to some relevant attribute, such as the square footage of a house, the number of bedrooms, the number of bathrooms, and the number of minutes (walking) to the center of town. Formally, they call one row in this dataset a feature vector, and the object (e.g. a house) is associated with an example. Feature vectors like this are essential for all the classic machine-learning problems. They'll typically denote the feature vector for any one example xi and the set of feature vectors for all our examples X. What makes a problem a regression is the outputs. Say that you're in the market for a new home, you might want to estimate the fair market value of a house, given some features like these. The target value, the price of sale, is a real number. They denote any individual target, Yi.

(Corresponding to example xi) and the set of all targets y (corresponding to all examples X). When our targets take on arbitrary real values in some range, they call this a regression problem. The goal of our model is to produce predictions (guess of the price, in our example) that closely approximate the actual target values.

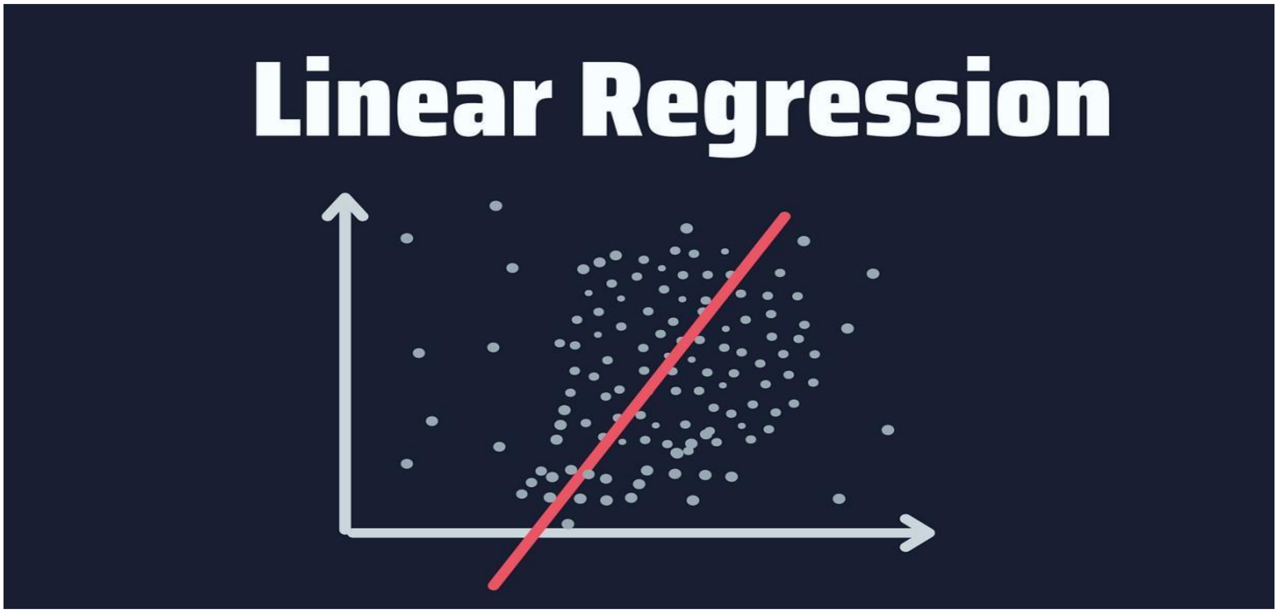


Figure 3 Linear regression

2.2.2

Unsupervised Learning :

Unsupervised learning is a machine learning technique in which models are not supervised using a training dataset. Instead, models themselves find the hidden patterns and insights from the given data. It can be compared to learning which takes place in the human brain while learning new things. It can be defined as:

Unsupervised learning is a type of machine learning in which models are trained using an unlabeled dataset and are allotted to act on that data without any supervision.

Unsupervised learning cannot be directly applied to a regression or classification problem because, unlike supervised learning, they have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of the dataset, group that data according to similarities, and represent that dataset in a compressed format. Example: Suppose the unsupervised learning algorithm is given an input dataset containing images of different types of cats and dogs. The algorithm is never trained upon the given dataset, which means it does not have any idea about the features of the dataset. The task of the unsupervised learning algorithm is to identify the image features on their own. An unsupervised learning algorithm will perform this task by clustering the image dataset into groups according to similarities between images.

Clustering.

The goal of unsupervised learning is to discover hidden patterns in any unlabeled data. One of the approaches to unsupervised learning is clustering. Clustering groups data points based on their similarities. Each group is called a cluster and contains data points with high similarity and low similarity with data points in other clusters. In short, the data points of a cluster are more similar to each other than they are to the data points of other clusters. The goal of clustering is to divide a set of data points in such a way that similar items fall into the same cluster, whereas dissimilar data points fall into different clusters. Clustering is crucial in multiple research fields in Bio Informatics such as analyzing unlabeled data which can be gene expression profiles, biomedical images, and so on. For example, clustering is often used in gene expression analysis to find groups of genes with similar expression patterns which may provide a useful understanding of gene functions and regulations, cellular processes, and so on.

There are many algorithms available for data clustering which use different ways to establish similarity between data points. The clustering algorithms can be broadly divided into many categories such as connectivity model, centroid model, density model, distribution model, group model, graph-based model, and so on.

Some of these are discussed below:

Connectivity model: This model assigns higher similarity to data points that are closer in one or multi-dimensional space than those points that are farther away. There are two approaches - first, it categorizes all data points into different clusters and then merges the data points about the distances among them. Second, it categorizes all data points into one single cluster and then partitions them into different clusters as the distance increases. This model is easy to understand but has problems in handling large datasets. One example is hierarchical clustering and its variants.

Centroid model: It is an iterative clustering algorithm in which similarity is based on the proximity of a data point to the centroids of the clusters. K-means clustering is one example of this model. It needs a few clusters before running and then divides data points into these many clusters iteratively. Therefore, to use K-Means, users should acquire some prior knowledge about the dataset.

Density model: This model searches one or multi-dimensional space for dense regions (having many data points in a small region). A popular example of a density model is DBSCAN (Density-Based Spatial Clustering of Applications with Noise).

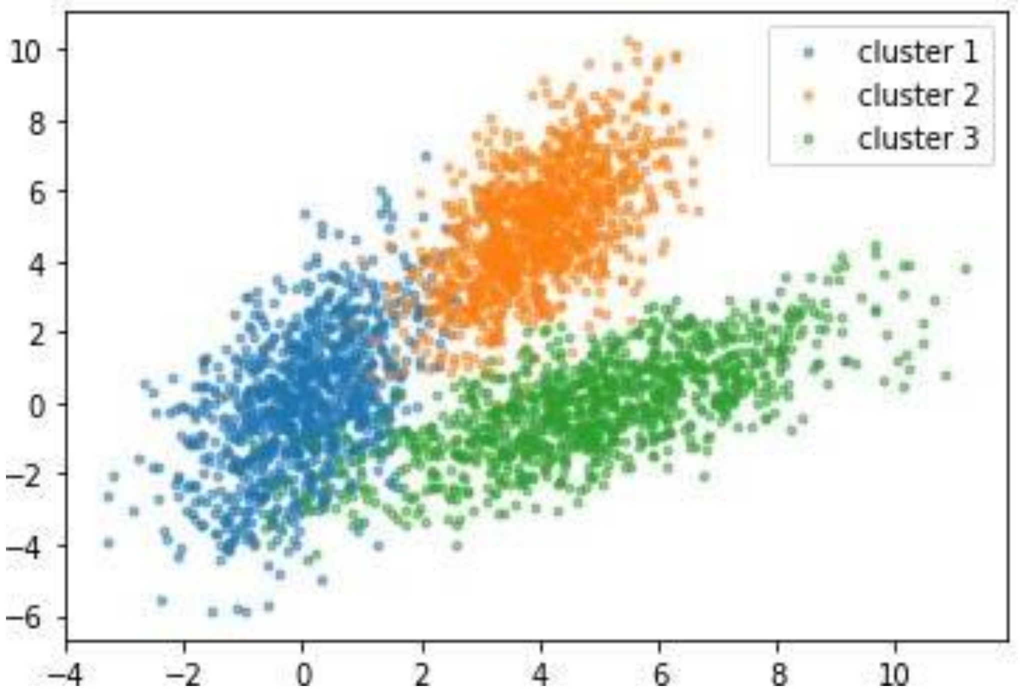


Figure 4 Clustering

2.2.3

Reinforcement Learning :

What is reinforcement learning?

Reinforcement learning is a machine learning training method based on rewarding desired behaviors and punishing undesired ones. In general, a reinforcement learning agent -- the entity being trained -- can perceive and interpret its environment, take actions, and learn through trial and error.

Reinforcement learning is one of several approaches developers use to train machine learning systems. What makes this approach important is that it empowers an agent, whether it's a feature in a video game or a robot in an industrial setting, to learn to navigate the complexities of the environment it was created. Over time, through a feedback system that typically includes rewards and punishments, the agent learns from its environment and optimizes its behaviors.

How does reinforcement learning work?

In reinforcement learning, developers devise a method of rewarding desired behaviors and punishing negative behaviors. This method assigns positive values to the desired actions to encourage the agent to use them, while negative values are assigned to undesired behaviors to discourage them. This programs the agent to seek long-term and maximum overall rewards to achieve an optimal solution.

These long-term goals help prevent the agent from getting stuck on less important goals. With time, the agent learns to avoid the negative and seek the positive. This learning method has been adopted in artificial intelligence (AI) as a way of directing unsupervised machine learning through rewards or positive reinforcement and penalties or negative reinforcement.

Applications and examples of reinforcement learning:

While reinforcement learning has been a topic of much interest in the field of AI, its widespread, real-world adoption and application remain limited. Noting this, however, research papers abound on theoretical applications, and there have been some successful use cases.

Current uses include but are not limited to the following:

- Gaming.
- Resource management.
- Personalized recommendations.
- Robotics.

Gaming is likely the most common use for reinforcement learning, as it can achieve superhuman performance in numerous games. A common example involves the game Pac-Man.

A learning algorithm playing Pac-Man might have the ability to move in one of four possible directions, barring obstruction. From pixel data, an agent might be given a numeric reward for the result of a unit of travel: 0 for empty spaces, 1 for pellets, 2 for fruit, 3 for power pellets, 4 for ghost post-power pellets, 5 for collecting all pellets to complete a level, and a 5-point deduction for collision with a ghost. The agent starts from randomized play and moves to more sophisticated play, learning the goal of getting all pellets to complete the level. Given time, an agent might even learn tactics such as conserving power pellets until needed for self-defense.

Reinforcement learning can operate in a situation as long as a clear reward can be applied. In enterprise resource management, reinforcement algorithms allocate limited resources to different tasks as long as there's an overall goal it is trying to achieve. A goal in this circumstance would be to save time or conserve resources.

In robotics, reinforcement learning has found its way into limited tests. This type of machine learning can provide robots with the ability to learn tasks a human teacher cannot demonstrate, to adapt a learned skill to a new task, and to achieve optimization even when analytic formulation isn't available.

Reinforcement learning is also used in operations research, information theory, game theory, control theory, simulation-based optimization, multi-agent systems, swarm intelligence, statistics, genetic algorithms, and ongoing Industrial automation efforts.

Challenges of applying reinforcement learning:

Reinforcement learning, while high in potential, comes with some tradeoffs. It can be difficult to deploy and remains limited in its application. One of the barriers to the deployment of this type of machine learning is its reliance on an exploration of the environment.

For example, if you were to deploy a robot that was reliant on reinforcement learning to navigate a complex physical environment, it would seek new states and take different actions as it moves. With this type of reinforcement learning problem, however, it's difficult to consistently take the best actions in a real-world environment because of how frequently the environment changes.

The time required to ensure the learning is done properly through this method can limit its usefulness and be intensive on computing resources. As the training environment grows more complex, so too do the demands on time and computing resources.

Supervised learning can deliver faster, more efficient results than reinforcement learning to companies if the proper amount of data is available, as it can be employed with fewer resources.

Common reinforcement learning algorithms:

Rather than referring to a specific algorithm, the field of reinforcement learning is made up of several algorithms that take somewhat different approaches.

The differences are mainly because of the different strategies they use to explore their environments:

State-action-reward-state-action. This reinforcement learning algorithm starts by giving the agent what's known as a policy. Determining the optimal policy-based approach requires looking at the probability of certain actions resulting in rewards, or beneficial states, to guide its decision-making.

Q-learning. This approach to reinforcement learning takes the opposite approach. The agent receives no policy and learns about an action's value based on exploration of its environment. This approach isn't model-based but instead is more self-directed. Real-world implementations of Q-learning are often written using Python programming.

Deep Q-networks. Combined with deep Q-learning, these algorithms use neural networks in addition to reinforcement learning techniques. They are also referred to as deep reinforcement learning and use reinforcement learning's self-directed environment exploration approach. As part of the learning process, these networks base future actions on a random sample of past beneficial actions.

How is reinforcement learning different from supervised and unsupervised learning?

Reinforcement learning is considered a branch of machine learning. However, it does have some similarities to other types of machine learning, which break down into the following four domains:

Supervised learning. In supervised learning, algorithms train on a body of labeled data. Supervised learning algorithms can only learn attributes that are specified in the data set. A common application of supervised learning is image recognition models. These models receive a set of labeled images and learn to distinguish common attributes of predefined forms.

Unsupervised learning. In unsupervised learning, developers turn algorithms loose on fully unlabeled data. The algorithms learn by cataloging their observations about data features without being told what to look for.

Semi-supervised learning. This method takes a middle-ground approach. Developers enter a relatively small set of labeled training data, as well as a larger corpus of unlabeled data. The algorithm is then instructed to extrapolate what it learns from the labeled data to the unlabeled data and draw conclusions from the set as a whole.

Reinforcement learning. This takes a different approach. It situates an agent in an environment with clear parameters defining beneficial activity and nonbeneficial activity and an overarching endgame to reach.

Reinforcement learning is similar to supervised learning in that developers must give algorithms specified goals and define reward functions and punishment functions. This means the level of explicit programming required is greater than in unsupervised learning. But, once these parameters are set, the algorithm operates on its own -- making it more self-directed than supervised learning algorithms. For this reason, people sometimes refer to reinforcement learning as a branch of semi-supervised learning; in truth, though, it is most often acknowledged as its type of machine learning.

2.3 Expert System:-

An expert system is a computer program designed to emulate and mimic the decision-making abilities of a human expert in a specific domain. It's a type of artificial intelligence (AI) system that utilizes knowledge stored in a knowledge base to reason and make decisions like a human expert.

Expert systems have been used in various fields such as medicine, finance, engineering, and manufacturing, among others. They are valuable tools for decision support, problem-solving, and knowledge management, especially in situations where access to human experts may be limited or costly.

Expert systems typically consist of two main components:

1. **Knowledge Base**: This component stores information, facts, rules, and heuristics about a particular domain. It's essentially the repository of knowledge that the system relies on to make decisions.

2. **Inference Engine**: The inference engine is the reasoning component of the expert system. It uses the knowledge stored in the knowledge base to conclude, make inferences, and provide solutions or recommendations based on the input provided by the user.

The inference engine uses one of two methods for acquiring information from the knowledge base:

Forward chaining reads and processes a set of facts to make a logical prediction about what will happen next. An example of forward chaining would be making predictions about the movement of the stock market.

Backward chaining reads and processes a set of facts to reach a logical conclusion about why something happened. An example of backward chaining would be examining a set of symptoms to reach a medical diagnosis.

An expert system relies on having a good knowledge base. Experts add information to the knowledge base, and nonexperts use the system to solve complex problems that would usually require a human expert.

The process of building and maintaining an expert system is called knowledge engineering. Knowledge engineers ensure that expert systems have all the necessary information to solve a problem. They use various knowledge representation methodologies, such as symbolic patterns, to do this. The system's capabilities can be enhanced by expanding the knowledge base or creating new sets of rules.

Applications and use cases of expert systems:

Expert systems can be effective in specific domains or subject areas where experts are required to make diagnoses, judgments, or predictions.

These systems have played a large role in many industries, including the following:

Financial services, where they make decisions about asset management, act as robo-advisors, and make predictions about the behavior of various markets and other financial indicators.

Mechanical engineering, where they troubleshoot complex electromechanical machinery.

Telecommunications, where are used to make decisions about network technologies used and maintenance of existing networks.

Healthcare, where they assist with medical diagnoses.

Agriculture, where they forecast crop damage.

Customer service, where they help schedule orders, route customer requests and solve problems.

Transportation, where they contribute in a range of areas, including pavement conditions, traffic light control, highway design, bus and train scheduling and maintenance, and aviation flight patterns and air traffic control.

Law, where automation is starting to be used to deliver legal services, make civil case evaluations and assess product liability.

What are some examples of expert systems?

Expert systems that are in use include the following examples:

Cadet (Cancer Decision Support Tool) is used to identify cancer in its earliest stages.

DENDRAL helps chemists identify unknown organic molecules.

DXplain is a clinical support system that diagnoses various diseases.

MYCIN identifies bacteria such as bacteremia and meningitis and recommends antibiotics and dosages.

PXDES determines the type and severity of lung cancer a person has.

R1/XCON is an early manufacturing expert system that automatically selects and orders computer components based on customer specifications.

What are the advantages of expert systems?

Expert systems have several benefits over the use of human experts:

Accuracy. Expert systems are not prone to human error or emotional influence. They make decisions based on defined rules and facts.

Permanence. Human experts eventually leave their role, and a lot of specific knowledge may go with them. Knowledge-based systems provide a permanent repository for knowledge and information.

Logical deduction. Expert systems conclude from existing facts using various types of rules, such as if-then rules.

Cost control. Expert systems are relatively inexpensive compared to the cost of employing human experts. They can help reach decisions more efficiently, which saves time and cuts costs.

Multiple experts. Multiple experts contribute to an expert system's knowledge base. This provides more knowledge to draw from and prevents any expert from skewing the decision-making.

What are the challenges of expert systems?

Among expert systems' shortcomings are the following:

Linear thinking. Expert systems lack true problem-solving ability. One of the advantages of human intelligence is that it can reason in nonlinear ways and use ancillary information to conclude.

Lack of intuition. Human intuition enables people to use common sense and gut feelings to solve problems. Machines don't have intuition. Emulating gut-feeling decision-making using mechanical logic could take much longer than an expert using intrinsic heuristic knowledge to come to a quick conclusion.

Lack of emotion. In some cases -- medical diagnoses, for example -- human emotion is useful and necessary. For example, the disclosure of sensitive medical information to a patient requires emotional intelligence that an expert system may not have.

Points of failure. Expert systems are only as good as the quality of their knowledge base. If they are supplied with inaccurate information, it can compromise their decisions.

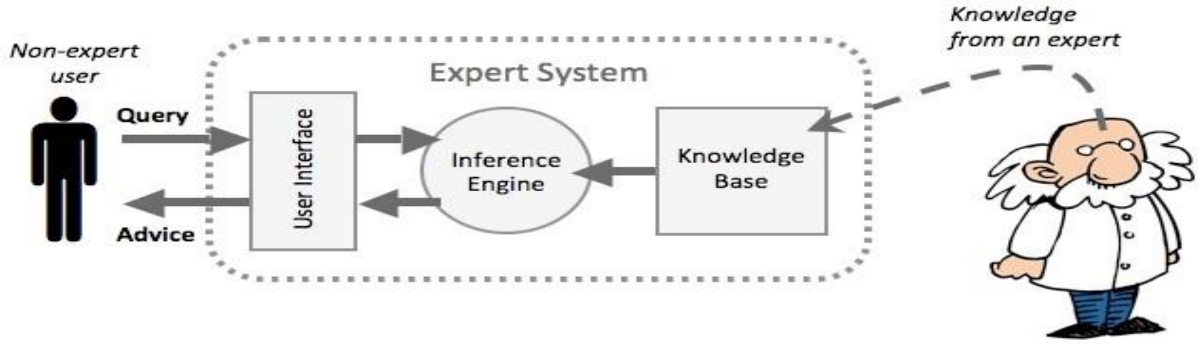


Figure 5 Expert system

Chapter 3

Literature Review

Paper 1:

1. Name article: Review of Hematology-Oncology Emergencies for Internal Medicine

Medicine Residents.

2. year:2023

3. Conclusions: In this review article, we covered the most common hematology hematology and oncology emergencies that internal medicine residents

can encounter during their practice. Unfortunately, the onset of cancer complications can be insidious and well ahead of symptoms caused by the tumor itself. Thus, proper diagnosis and treatment can be delayed, potentially leading to death or significant morbidity. This review covers several oncology and hematology emergencies and will be helpful to early internal medicine trainees to get familiar with diagnostic and treatment modalities.

Paper 2:

1. Name article: Graves' Ophthalmopathy: Epidemiology and Natural History.

2. year:2014

3. Conclusions: GO is an autoimmune disease of the orbit that is frequently associated with Graves' disease. The age-adjusted annual incidence of clinically relevant GO in a representative county is estimated to be 16 per 100,000 100,000 population in women and 2.9 per 100,000 in men. The estimated prevalence of clinically relevant GO ranges from 0.1% to 0.3%. At the onset of onset of orbitopathy, 80-90% of GO patients have hyperthyroidism, with the rest

the rest having either euthyroidism or hypothyroidism. Approximately 5% of patients of patients exhibit late reactivation of GO.

Paper 3:

1. Name article: Diagnosis of Gallbladder Tumors.

2.year:2011

3. Conclusions: Mass screening by using ultrasonography can detect gallbladder abnormalities, among which early gallbladder carcinoma should be distinguished from other diseases. Further evaluation using endoscopic ultrasonography is useful, but recent advances in contrast-enhanced ultrasonography are expected to increase capabilities for early diagnosis.

Paper 4:

1. Name article: Prevention of Gastric Cancer by Helicobacter pylori Eradication

2. year:2010

3. Conclusions: Whether or not H. pylori eradication therapy can suppress gastric cancer has been a serious issue in this field. New evidence obtained from our study in Japan may be helpful for the prevention and eventual elimination of gastric cancer.

Paper 5:

1. Name article: Performance of Cardiovascular Physical Exam Skills by Internal Medicine Residents

2. year:2024

3. Conclusions: Internal medicine interns had variable skills in performing and interpreting the cardiovascular physical exam. Improving cardiovascular exam skills would likely lead to increased identification of relevant cardiovascular findings, inform clinical decision-making, and improve overall patient care.

Paper 6:

1. Name article: Attitudes of Internal Medicine Nurses, Surgical Nurses and Midwives towards Reporting of Clinical Adverse Events

2. year:2024

3. Conclusions: Based on the research material presented and the analysis of the results, the following can be concluded:

- the attitudes of the internal medicine nurses, surgical nurses, and midwives towards reporting clinical adverse events in P-RoCAES were positive, especially those with a master’s degree,
- respondents with a master’s degree had the best understanding of the importance of the criteria for identifying incidents and they also had the highest perception of the expectations of colleagues and the benefits of reporting adverse events (P-RoCAES subscales),
- in terms of guilt (P-RoCAES subscale), the longer employment tenure and having a master’s degree were associated with a higher sense of guilt and also the internal medicine nurses had higher guilt feelings compared to midwives and surgical nurses,
- the longer respondents worked, the more positive their attitude became towards the importance of the practice of transparency in procedures, but there was a negative attitude towards the benefits of adverse event reporting (P-RoCAES subscales).

Paper 7:

1. Name article: American Board of Internal Medicine Nephrology Procedure Requirements for Initial Certification: Time for a Change and Pursuing Consensus in the Nephrology Community

2. year:2021

3. Conclusions: In March 2021, the ABIM Nephrology Board embarked on this policy journey to revise the procedure requirements for nephrology certification. The approved change to an Opportunity to Train standard for temporary catheters and kidney biopsies better reflects the current state of training in most fellowships, but the board expects that programs will retain training opportunities for interested fellows who anticipate joining employment settings where these competencies are required.

Paper 8:

1. Name article: Profile, reasons for hospitalization and nursing diagnoses of refugee-native patients admitted to internal medicine clinic-an evaluation from a nursing perspective

2. year:2024

3. Conclusions: As a result of the study, it was observed that the duration of hospitalization, reasons for hospitalization, and nursing diagnoses of local and refugee patients were similar. In addition, it was determined that the patient's medical hospitalization diagnoses and nursing diagnoses were compatible. According to this result, it is thought that nursing diagnoses and care are made by considering cultural differences and medical diagnoses of patients.

Paper 9:

1. Name article: Training, perception, and experience of the point of care ultrasound amongst internal medicine trainees across Southwest England

2. year:2024

3. Conclusions: Point-of-care ultrasound has several applications including being used as an important diagnostic tool and as a key cornerstone to ensuring adequate site selection for invasive medical procedures. Despite the multitude of POCUS applications, few IMT trainees are obtaining sufficient experience or training in POCUS. IMTs perceive POCUS training as highly valuable and would relish the opportunity for an enhanced POCUS training program to be delivered.

Paper 10:

1. Name article: The Utility of *Capsicum annuum* L. in Internal Medicine and Dentistry: A Comprehensive Review.

2. year:2022

3. Conclusions: Since ancient times, Capsaicin has been used for its particular beneficial properties. However, only recently have its mechanisms of action and potentialities been highlighted. In addition to the healthy properties already studied (namely, the improvement of dyslipidemia, the beneficial effects in chronic kidney disease, rather than the improvement of insulin sensitivity or the reduction in body weight in obese subjects, etc.), Capsaicin would seem to be a valid therapeutic option for the treatment of PTTN.

In this regard, we present our experience in the local use of Capsaicin in the treatment of PTTN. Initially, in our study population, Capsaicin's contact with oral mucosa caused hyperalgesia and increased neuropathic pain in some patients. However, its continuative application desensitized neurosensitive fibers, showing pain relief.

In conclusion, Capsaicin seems to be well-tolerated by patients, despite the local burning pain and other minor local side effects observed during the first applications, making it a valid new therapeutic agent, free from side effects, not only in internal medicine but also in dentistry.

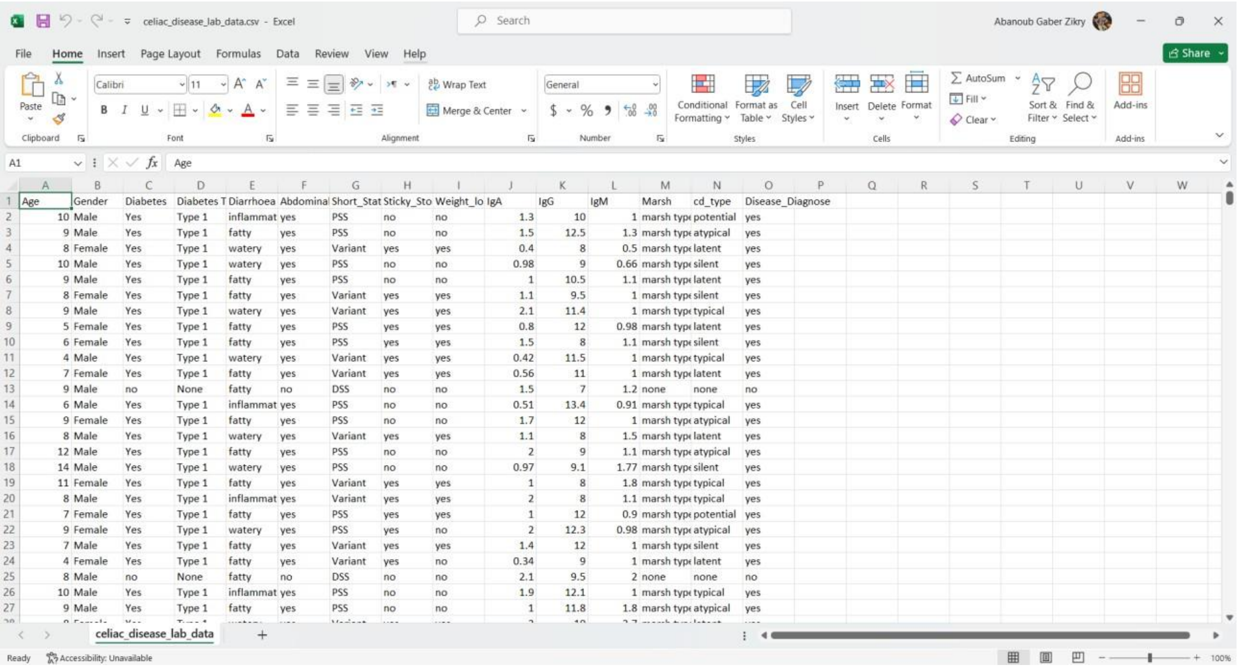
Chapter 4

Implementation

➢ The first stage:

4.1 Data collection:-

We got the first data set from Kaggle.



1	Age	Gender	Diabetes	Diabetes T	Diarrhoea	Abdomina	Short_Stat	Sticky_Sto	Weight_lo	IgA	IgG	IgM	Marsh	cd_type	Disease_Diagnose
2	10	Male	Yes	Type 1	inflammat	yes	PSS	no	no	1.3	10	1	marsh typ	potential	yes
3	9	Male	Yes	Type 1	fatty	yes	PSS	no	no	1.5	12.5	1.3	marsh typ	atypical	yes
4	8	Female	Yes	Type 1	watery	yes	Variant	yes	yes	0.4	8	0.5	marsh typ	latent	yes
5	10	Male	Yes	Type 1	watery	yes	PSS	no	no	0.98	9	0.66	marsh typ	silent	yes
6	9	Male	Yes	Type 1	fatty	yes	PSS	no	no	1	10.5	1.1	marsh typ	latent	yes
7	8	Female	Yes	Type 1	fatty	yes	Variant	yes	yes	1.1	9.5	1	marsh typ	silent	yes
8	9	Male	Yes	Type 1	watery	yes	Variant	yes	yes	2.1	11.4	1	marsh typ	typical	yes
9	5	Female	Yes	Type 1	fatty	yes	PSS	yes	yes	0.8	12	0.98	marsh typ	latent	yes
10	6	Female	Yes	Type 1	fatty	yes	PSS	yes	yes	1.5	8	1.1	marsh typ	silent	yes
11	4	Male	Yes	Type 1	watery	yes	Variant	yes	yes	0.42	11.5	1	marsh typ	typical	yes
12	7	Female	Yes	Type 1	fatty	yes	Variant	yes	yes	0.56	11	1	marsh typ	latent	yes
13	9	Male	no	None	fatty	no	DSS	no	no	1.5	7	1.2	none	none	no
14	6	Male	Yes	Type 1	inflammat	yes	PSS	no	no	0.51	13.4	0.91	marsh typ	typical	yes
15	9	Female	Yes	Type 1	fatty	yes	PSS	no	no	1.7	12	1	marsh typ	atypical	yes
16	8	Male	Yes	Type 1	watery	yes	Variant	yes	yes	1.1	8	1.5	marsh typ	latent	yes
17	12	Male	Yes	Type 1	fatty	yes	PSS	no	no	2	9	1.1	marsh typ	atypical	yes
18	14	Male	Yes	Type 1	watery	yes	PSS	no	no	0.97	9.1	1.77	marsh typ	silent	yes
19	11	Female	Yes	Type 1	fatty	yes	Variant	yes	yes	1	8	1.8	marsh typ	typical	yes
20	8	Male	Yes	Type 1	inflammat	yes	Variant	yes	yes	2	8	1.1	marsh typ	typical	yes
21	7	Female	Yes	Type 1	fatty	yes	PSS	yes	yes	1	12	0.9	marsh typ	potential	yes
22	9	Female	Yes	Type 1	watery	yes	PSS	yes	no	2	12.3	0.98	marsh typ	atypical	yes
23	7	Male	Yes	Type 1	fatty	yes	Variant	yes	yes	1.4	12	1	marsh typ	silent	yes
24	4	Female	Yes	Type 1	fatty	yes	Variant	yes	no	0.34	9	1	marsh typ	latent	yes
25	8	Male	no	None	fatty	no	DSS	no	no	2.1	9.5	2	none	none	no
26	10	Male	Yes	Type 1	inflammat	yes	PSS	no	no	1.9	12.1	1	marsh typ	typical	yes
27	9	Male	Yes	Type 1	fatty	yes	PSS	no	no	1	11.8	1.8	marsh typ	atypical	yes

Figure c old data set

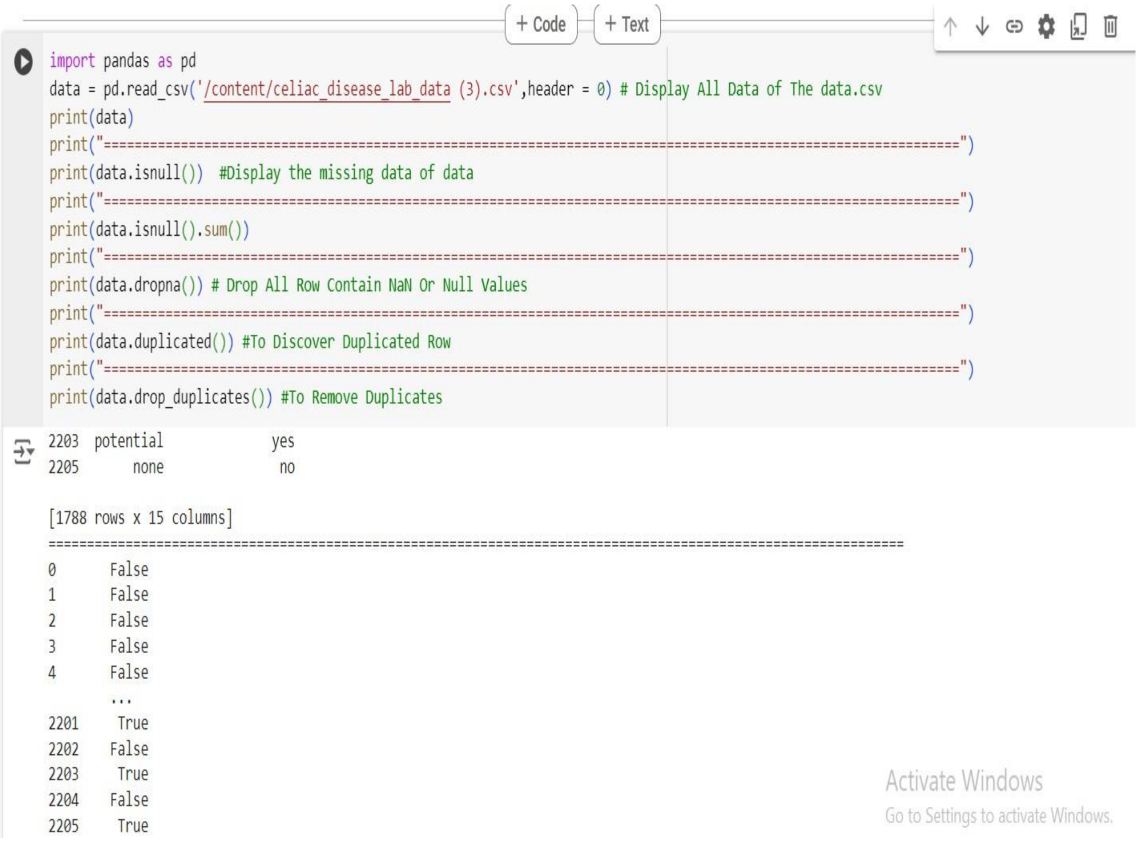
This data set is about celiac disease.

This data contains one disease, which is celiac disease, as it contains 15 columns, 14 of which contain the symptoms that appear on the sick person, including gender, age, sugar level, and other information that the patient gives to the chat, and then the chat responds to him if he is infected with this disease or not. The outfit is contained in the last column.

4.2 Data Preprocessing:-

Preprocessing is a crucial step in data analysis, machine learning, and other computational tasks. It involves preparing raw data for analysis by cleaning, transforming, and organizing it into a suitable format. Here are the key tasks involved in preprocessing:

- Data Cleaning
- Data Transformation
- Data Integration
- Data Reduction
- Data Discretization
- Data Imputation
- Handling Outliers
- Data Augmentation
- Data Splitting



```python
import pandas as pd
data = pd.read_csv('/content/celiac_disease_lab_data (3).csv',header = 0) # Display All Data of The data.csv
print(data)
print("==============================================================================================================")
print(data.isnull())  #Display the missing data of data
print("==============================================================================================================")
print(data.isnull().sum())
print("==============================================================================================================")
print(data.dropna()) # Drop All Row Contain NaN Or Null Values
print("==============================================================================================================")
print(data.duplicated()) #To Discover Duplicated Row
print("==============================================================================================================")
print(data.drop_duplicates()) #To Remove Duplicates
```

```
2203  potential              yes
2205       none               no

[1788 rows x 15 columns]
==============================================================================================================
0       False
1       False
2       False
3       False
4       False
        ...
2201     True
2202    False
2203     True
2204    False
2205     True
```

Figure 7 Cleaning old data set

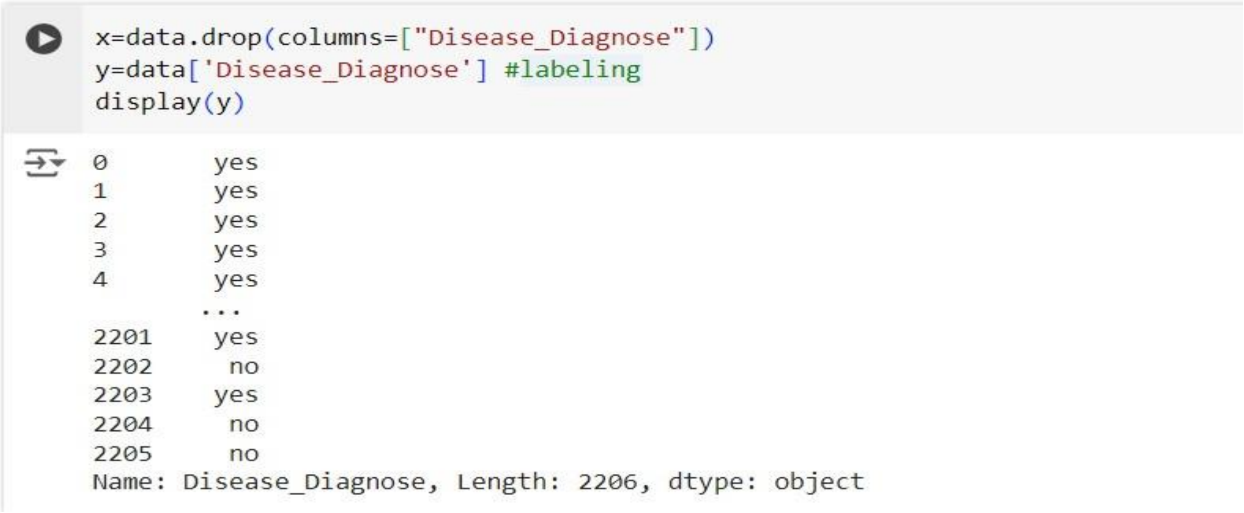
1- We use is null() function to detect missing values in diseases or symptoms.

2- We use the drop () function to remove the row that contains Null values.

3- We use duplicated() function to discover duplicated row.

4- We use drop_duplicates() function to remove duplicates.

We did this to give us a better performance



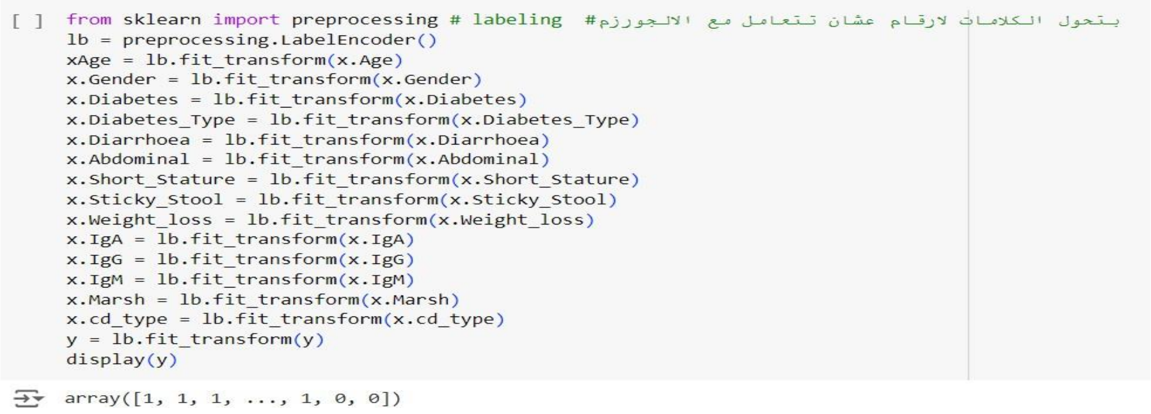
```
x=data.drop(columns=["Disease_Diagnose"])
y=data['Disease_Diagnose'] #labeling
display(y)
```

```
0       yes
1       yes
2       yes
3       yes
4       yes
       ...
2201    yes
2202     no
2203    yes
2204     no
2205     no
Name: Disease_Diagnose, Length: 2206, dtype: object
```

Figure 8 Divides old data

We divided the data into (x,y) where X is symptoms and Y is a disease (labeling),

Because we give the model X and it gives us Y.



```
[ ]  from sklearn import preprocessing # labeling  #بتحول الكلامات لارقام عشان تتعامل مع الالجورزم
     lb = preprocessing.LabelEncoder()
     xAge = lb.fit_transform(x.Age)
     x.Gender = lb.fit_transform(x.Gender)
     x.Diabetes = lb.fit_transform(x.Diabetes)
     x.Diabetes_Type = lb.fit_transform(x.Diabetes_Type)
     x.Diarrhoea = lb.fit_transform(x.Diarrhoea)
     x.Abdominal = lb.fit_transform(x.Abdominal)
     x.Short_Stature = lb.fit_transform(x.Short_Stature)
     x.Sticky_Stool = lb.fit_transform(x.Sticky_Stool)
     x.Weight_loss = lb.fit_transform(x.Weight_loss)
     x.IgA = lb.fit_transform(x.IgA)
     x.IgG = lb.fit_transform(x.IgG)
     x.IgM = lb.fit_transform(x.IgM)
     x.Marsh = lb.fit_transform(x.Marsh)
     x.cd_type = lb.fit_transform(x.cd_type)
     y = lb.fit_transform(y)
     display(y)
```

```
array([1, 1, 1, ..., 1, 0, 0])
```

Figure S Convert old data

we use the .fit _transform() function to convert the data string to (1,0)

4.3 Choosing a model:-

1-Random forest:

Random forests, typically used for structured data, can be adapted for text data by incorporating specific feature extraction and engineering techniques.

Random forests are a versatile and widely-used machine learning algorithm, applicable to a variety of tasks due to their robustness and ability to handle different types of data. Here are the primary uses of random forests:

1- Classification:

Binary Classification: Distinguishing between two classes (e.g., spam vs. not spam, disease vs. no disease).

Multi-Class Classification: Distinguishing among more than two classes (e.g., classifying types of flowers, recognizing handwriting digits).

2- Regression:

Predicting Continuous Values: Used for tasks where the output is a continuous value (e.g., predicting house prices, stock prices, temperature).

3- Feature Selection:

Identifying Important Features: Random forests can rank features by their importance, helping in feature selection and dimensionality reduction.

4- Anomaly Detection:

Outlier Detection: Identifying unusual or anomalous data points that do not fit the general pattern of the dataset.

5- Imputation of Missing Values:

Filling Missing Data: Using the majority vote or the average of trees to predict and fill missing values in the dataset.

6- Clustering:

Unsupervised Learning: Though less common, random forests can be used in clustering by using the proximity matrix to group similar data points.

2- Support vector machines (SVM):-

Support Vector Machines (SVM) are a versatile and powerful machine learning algorithm widely used for both classification and regression tasks. Here are some common applications and use cases of SVM:

1. Classification:

Binary Classification: SVM is frequently used for tasks where the objective is to distinguish between two classes. Examples include:

Spam Detection: Classifying emails as spam or not spam.

Disease Diagnosis: Predicting whether a patient has a certain disease (e.g., cancer vs. no cancer).

Sentiment Analysis: Determining if a review or social media post is positive or negative.

Multi-Class Classification: Although SVM is inherently a binary classifier, it can be extended to handle multiple classes through techniques like one-vs-one and one-vs-rest. Examples include:

Handwriting Recognition: Classifying handwritten digits (e.g., digits 0-9).

Image Classification: Categorizing images into different classes (e.g., identifying different species of animals).

2. Regression:

Support Vector Regression (SVR): SVM can also be used for regression tasks where the goal is to predict a continuous value. Examples include:

Stock Price Prediction: Predicting future stock prices based on historical data.

Housing Price Prediction: Estimating the price of a house based on features like location, size, and amenities.

3. Text and NLP Applications:

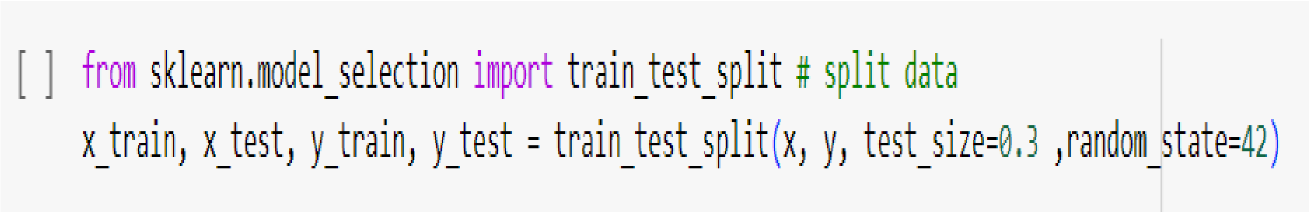
Text Classification: Categorizing text into predefined categories. Examples include:

News Categorization: Classifying news articles into topics like sports, politics, or entertainment.

Document Classification: Sorting documents into categories such as legal, medical, or financial.

Sentiment Analysis: Analyzing the sentiment expressed in text (e.g., determining if a tweet is positive, negative, or neutral).

4.4 Split training and testing:-



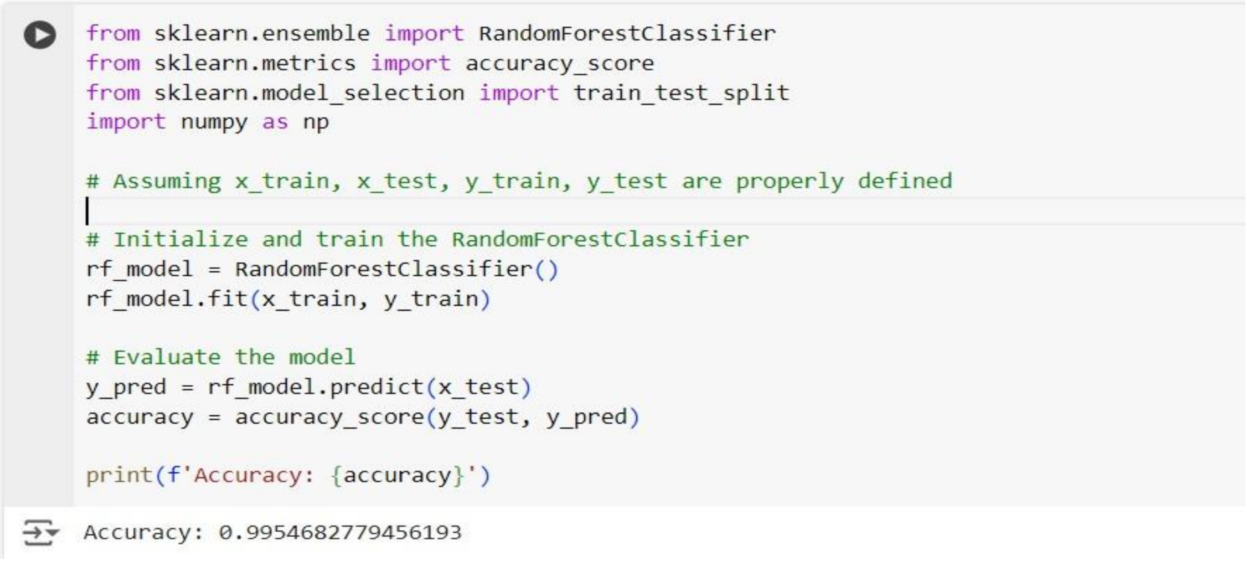
```
[ ] from sklearn.model_selection import train_test_split # split data
    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3 ,random_state=42)
```

Figure 10 Split old data

Split the train and test at the rate of train 70% and test model 30% to train the model and then test it

4.4.1Building model:-

1- Random Forest:

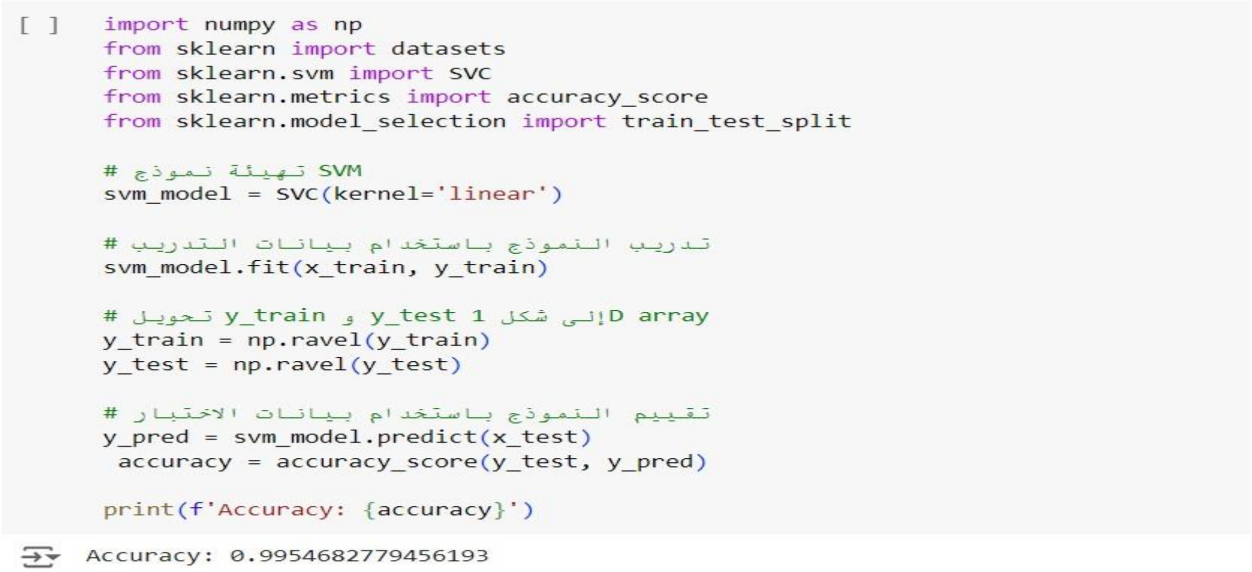


```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
import numpy as np

# Assuming x_train, x_test, y_train, y_test are properly defined

# Initialize and train the RandomForestClassifier
rf_model = RandomForestClassifier()
rf_model.fit(x_train, y_train)

# Evaluate the model
y_pred = rf_model.predict(x_test)
accuracy = accuracy_score(y_test, y_pred)

print(f'Accuracy: {accuracy}')
```

```
Accuracy: 0.9954682779456193
```

Figure 10 Train random forest on old data

2- Svm:



```
import numpy as np
from sklearn import datasets
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

# تهيئة نموذج SVM
svm_model = SVC(kernel='linear')

# تدريب النموذج باستخدام بيانات التدريب
svm_model.fit(x_train, y_train)

# تحويل y_train و y_test إلى شكل 1D array
y_train = np.ravel(y_train)
y_test = np.ravel(y_test)

# تقييم النموذج باستخدام بيانات الاختبار
y_pred = svm_model.predict(x_test)
 accuracy = accuracy_score(y_test, y_pred)

print(f'Accuracy: {accuracy}')
```

```
Accuracy: 0.9954682779456193
```

Figure 11 Train Svm on old data

4.4.2 Training model:-

1-We use .fit() function to train the model on (x_train , y_train)

2-We use .ravel() function to convert 1D array

3- We use .predict() function to test the model

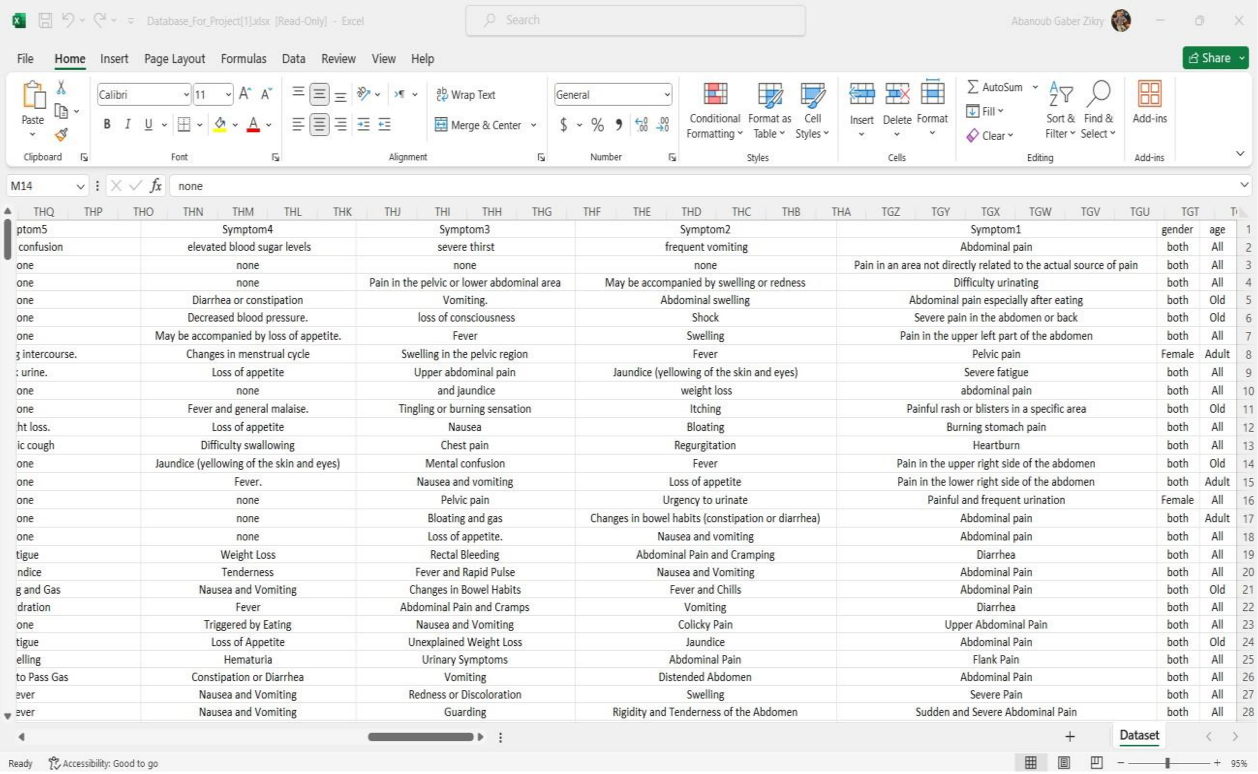
4.4.3 Evaluation model:-

We use accuracy_score() to evaluate the model to see if it is accurate or not.

➢ The Second stage:

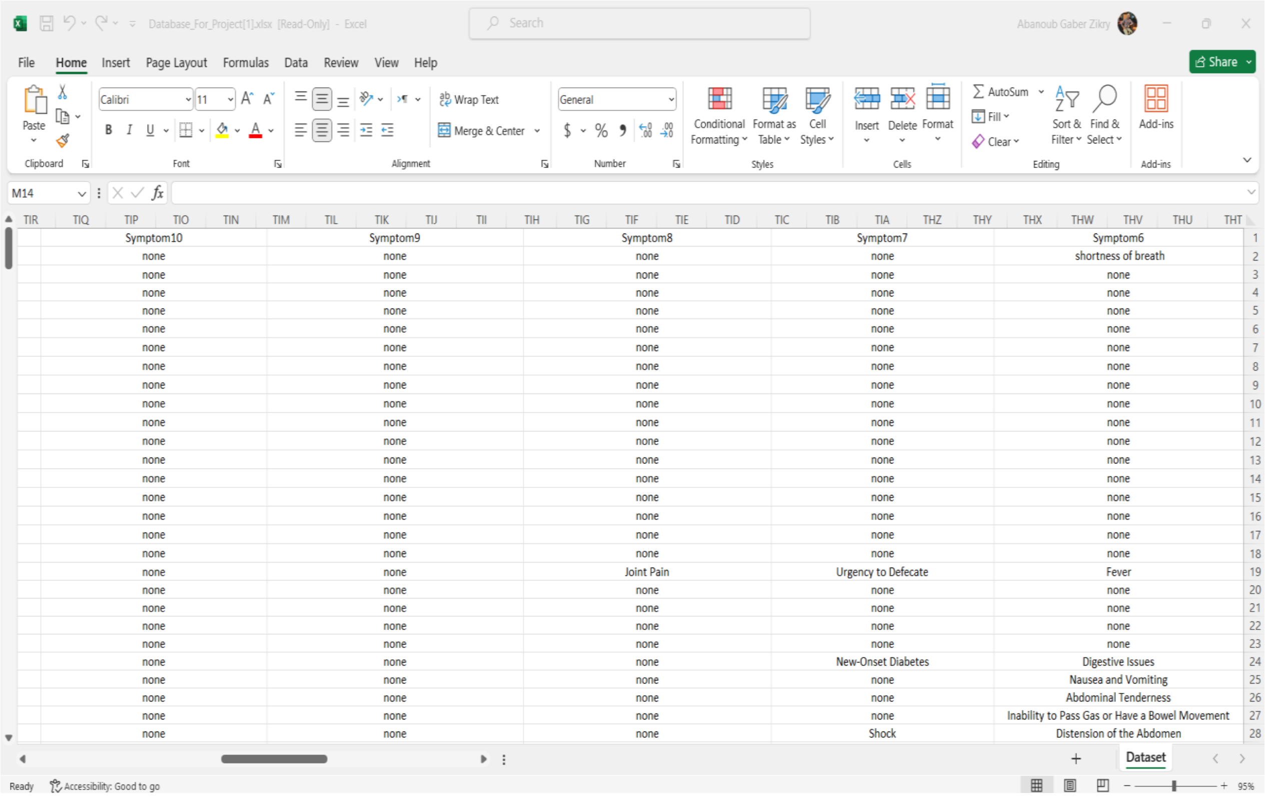
4.1 Data collection:-

At this stage, we developed the idea that we would collect data sets from doctors about all diseases of the internal medicine specialty, instead of ready-made data sets that contain one disease, so we worked on the data sets that contained one disease until we completed collecting this data that contains most of the diseases. Internal and most purposes, it contains 16 columns, including 12 columns for symptoms, a column for age, a column for gender, a column for the disease, and a column to describe the disease, so that the patient gives the chat the symptoms he feels, and the chat responds to him with the disease, its description, age, and gender.

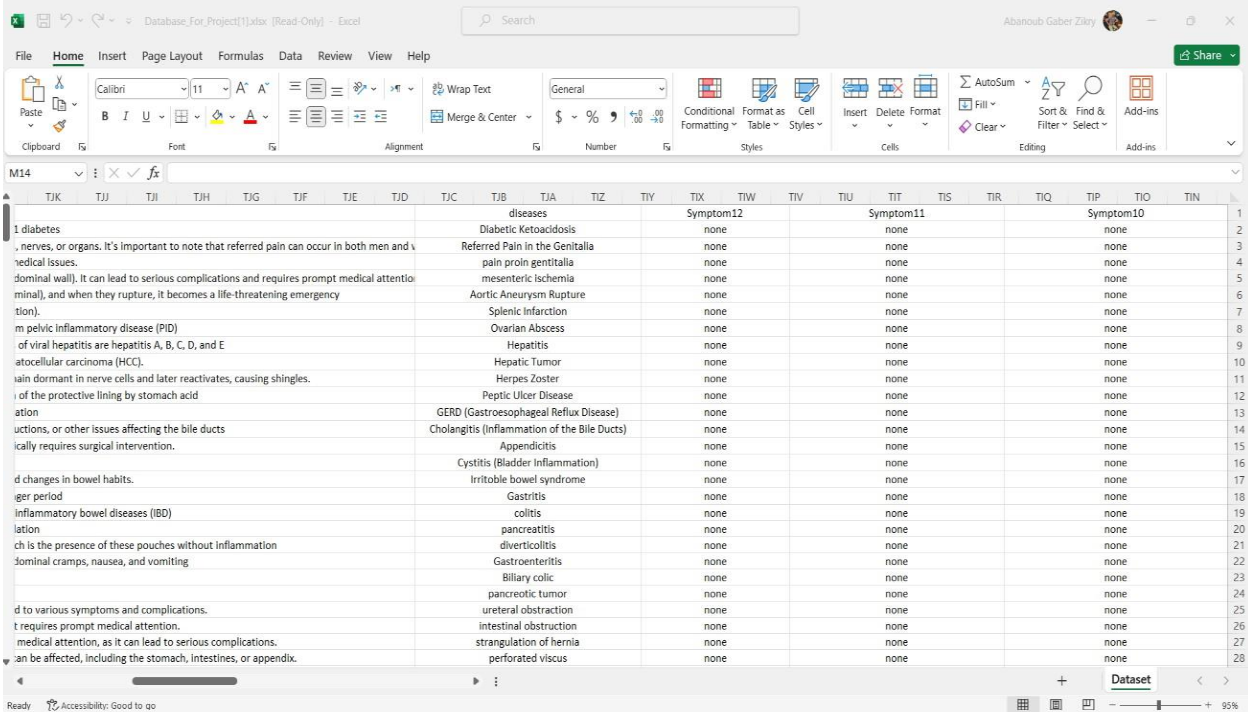


ptom5	Symptom4	Symptom3	Symptom2	Symptom1	gender	age	1
confusion	elevated blood sugar levels	severe thirst	frequent vomiting	Abdominal pain	both	All	2
one	none	none	none	Pain in an area not directly related to the actual source of pain	both	All	3
one	none	Pain in the pelvic or lower abdominal area	May be accompanied by swelling or redness	Difficulty urinating	both	All	4
one	Diarrhea or constipation	Vomiting.	Abdominal swelling	Abdominal pain especially after eating	both	Old	5
one	Decreased blood pressure.	loss of consciousness	Shock	Severe pain in the abdomen or back	both	Old	6
one	May be accompanied by loss of appetite.	Fever	Swelling	Pain in the upper left part of the abdomen	both	All	7
; intercourse.	Changes in menstrual cycle	Swelling in the pelvic region	Fever	Pelvic pain	Female	Adult	8
: urine.	Loss of appetite	Upper abdominal pain	Jaundice (yellowing of the skin and eyes)	Severe fatigue	both	All	9
one	none	and jaundice	weight loss	abdominal pain	both	All	10
one	Fever and general malaise.	Tingling or burning sensation	Itching	Painful rash or blisters in a specific area	both	Old	11
ht loss.	Loss of appetite	Nausea	Bloating	Burning stomach pain	both	All	12
ic cough	Difficulty swallowing	Chest pain	Regurgitation	Heartburn	both	All	13
one	Jaundice (yellowing of the skin and eyes)	Mental confusion	Fever	Pain in the upper right side of the abdomen	both	Old	14
one	Fever.	Nausea and vomiting	Loss of appetite	Pain in the lower right side of the abdomen	both	Adult	15
one	none	Pelvic pain	Urgency to urinate	Painful and frequent urination	Female	All	16
one	none	Bloating and gas	Changes in bowel habits (constipation or diarrhea)	Abdominal pain	both	Adult	17
one	none	Loss of appetite.	Nausea and vomiting	Abdominal pain	both	All	18
tigue	Weight Loss	Rectal Bleeding	Abdominal Pain and Cramping	Diarrhea	both	All	19
ndice	Tenderness	Fever and Rapid Pulse	Nausea and Vomiting	Abdominal Pain	both	All	20
g and Gas	Nausea and Vomiting	Changes in Bowel Habits	Fever and Chills	Abdominal Pain	both	Old	21
dration	Fever	Abdominal Pain and Cramps	Vomiting	Diarrhea	both	All	22
one	Triggered by Eating	Nausea and Vomiting	Colicky Pain	Upper Abdominal Pain	both	All	23
tigue	Loss of Appetite	Unexplained Weight Loss	Jaundice	Abdominal Pain	both	Old	24
elling	Hematuria	Urinary Symptoms	Abdominal Pain	Flank Pain	both	All	25
to Pass Gas	Constipation or Diarrhea	Vomiting	Distended Abdomen	Abdominal Pain	both	All	26
ever	Nausea and Vomiting	Redness or Discoloration	Swelling	Severe Pain	both	All	27
ever	Nausea and Vomiting	Guarding	Rigidity and Tenderness of the Abdomen	Sudden and Severe Abdominal Pain	both	All	28

Figure 12 New data set



Symptom10	Symptom9	Symptom8	Symptom7	Symptom6	1
none	none	none	none	shortness of breath	2
none	none	none	none	none	3
none	none	none	none	none	4
none	none	none	none	none	5
none	none	none	none	none	6
none	none	none	none	none	7
none	none	none	none	none	8
none	none	none	none	none	9
none	none	none	none	none	10
none	none	none	none	none	11
none	none	none	none	none	12
none	none	none	none	none	13
none	none	none	none	none	14
none	none	none	none	none	15
none	none	none	none	none	16
none	none	none	none	none	17
none	none	none	none	none	18
none	none	Joint Pain	Urgency to Defecate	Fever	19
none	none	none	none	none	20
none	none	none	none	none	21
none	none	none	none	none	22
none	none	none	none	none	23
none	none	none	New-Onset Diabetes	Digestive Issues	24
none	none	none	none	Nausea and Vomiting	25
none	none	none	none	Abdominal Tenderness	26
none	none	none	none	Inability to Pass Gas or Have a Bowel Movement	27
none	none	none	Shock	Distension of the Abdomen	28

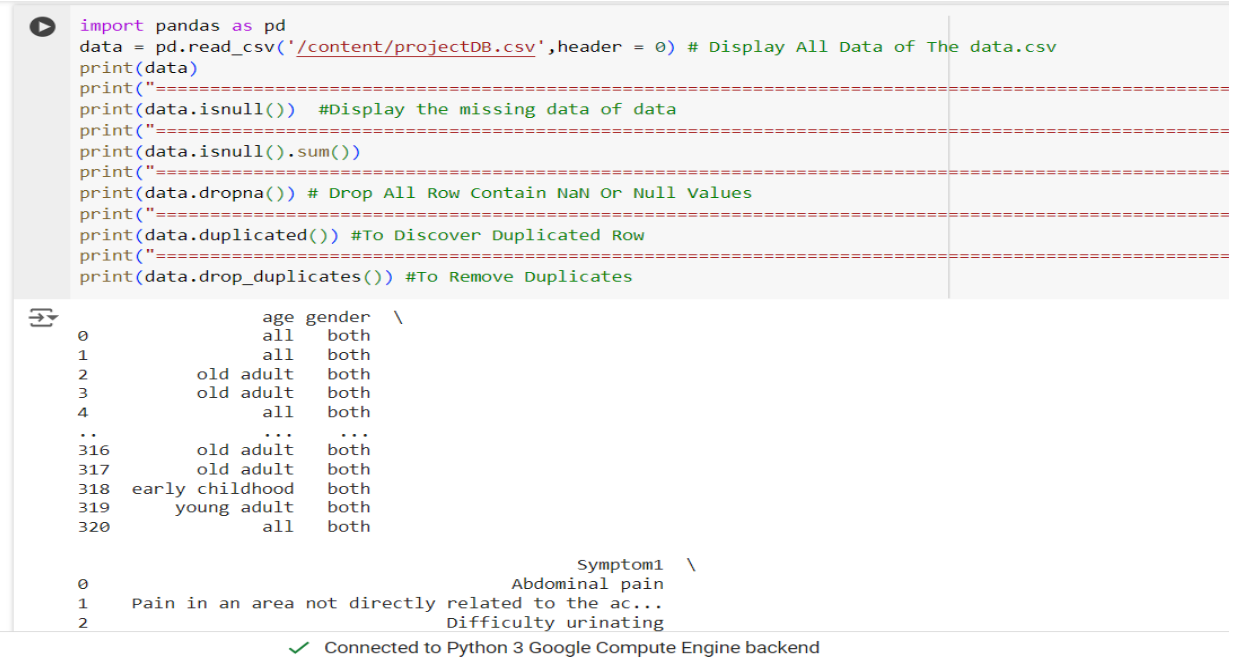


	diseases	Symptom12	Symptom11	Symptom10	1
1 diabetes	Diabetic Ketoacidosis	none	none	none	2
, nerves, or organs. It's important to note that referred pain can occur in both men and v	Referred Pain in the Genitalia	none	none	none	3
nedical issues.	pain proin gentitalia	none	none	none	4
dominal wall). It can lead to serious complications and requires prompt medical attentio	mesenteric ischemia	none	none	none	5
minal), and when they rupture, it becomes a life-threatening emergency	Aortic Aneurysm Rupture	none	none	none	6
tion).	Splenic Infarction	none	none	none	7
m pelvic inflammatory disease (PID)	Ovarian Abscess	none	none	none	8
of viral hepatitis are hepatitis A, B, C, D, and E	Hepatitis	none	none	none	9
atocellular carcinoma (HCC).	Hepatic Tumor	none	none	none	10
ain dormant in nerve cells and later reactivates, causing shingles.	Herpes Zoster	none	none	none	11
of the protective lining by stomach acid	Peptic Ulcer Disease	none	none	none	12
ation	GERD (Gastroesophageal Reflux Disease)	none	none	none	13
uctions, or other issues affecting the bile ducts	Cholangitis (Inflammation of the Bile Ducts)	none	none	none	14
ically requires surgical intervention.	Appendicitis	none	none	none	15
	Cystitis (Bladder Inflammation)	none	none	none	16
d changes in bowel habits.	Irritoble bowel syndrome	none	none	none	17
ger period	Gastritis	none	none	none	18
inflammatory bowel diseases (IBD)	colitis	none	none	none	19
lation	pancreatitis	none	none	none	20
ch is the presence of these pouches without inflammation	diverticolitis	none	none	none	21
dominal cramps, nausea, and vomiting	Gastroenteritis	none	none	none	22
	Biliary colic	none	none	none	23
	pancreotic tumor	none	none	none	24
d to various symptoms and complications.	ureteral obstraction	none	none	none	25
t requires prompt medical attention.	intestinal obstruction	none	none	none	26
medical attention, as it can lead to serious complications.	strangulation of hernia	none	none	none	27
an be affected, including the stomach, intestines, or appendix.	perforated viscus	none	none	none	28

Figure 13 New data set

This data set is about all diseases of the internal medicine specialty.

- ❖ **The data cleaning and preprocessing are the same as what we did in the previous data set.**

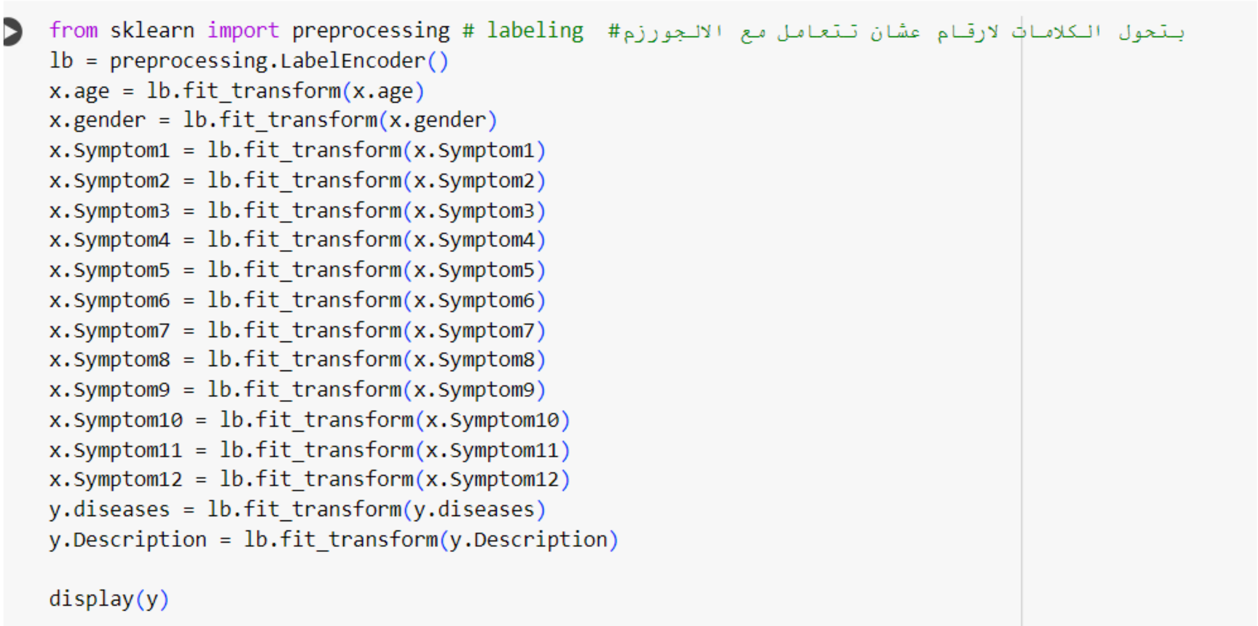


```
import pandas as pd
data = pd.read_csv('/content/projectDB.csv',header = 0) # Display All Data of The data.csv
print(data)
print("==========================================================================================
print(data.isnull())  #Display the missing data of data
print("==========================================================================================
print(data.isnull().sum())
print("==========================================================================================
print(data.dropna()) # Drop All Row Contain NaN Or Null Values
print("==========================================================================================
print(data.duplicated()) #To Discover Duplicated Row
print("==========================================================================================
print(data.drop_duplicates()) #To Remove Duplicates
```

```
                 age gender  \
0                all   both
1                all   both
2          old adult   both
3          old adult   both
4                all   both
..               ...    ...
316        old adult   both
317        old adult   both
318  early childhood   both
319      young adult   both
320              all   both

                                              Symptom1  \
0                                       Abdominal pain
1    Pain in an area not directly related to the ac...
2                                  Difficulty urinating
```

Connected to Python 3 Google Compute Engine backend

Figure 14 Cleaning new data set



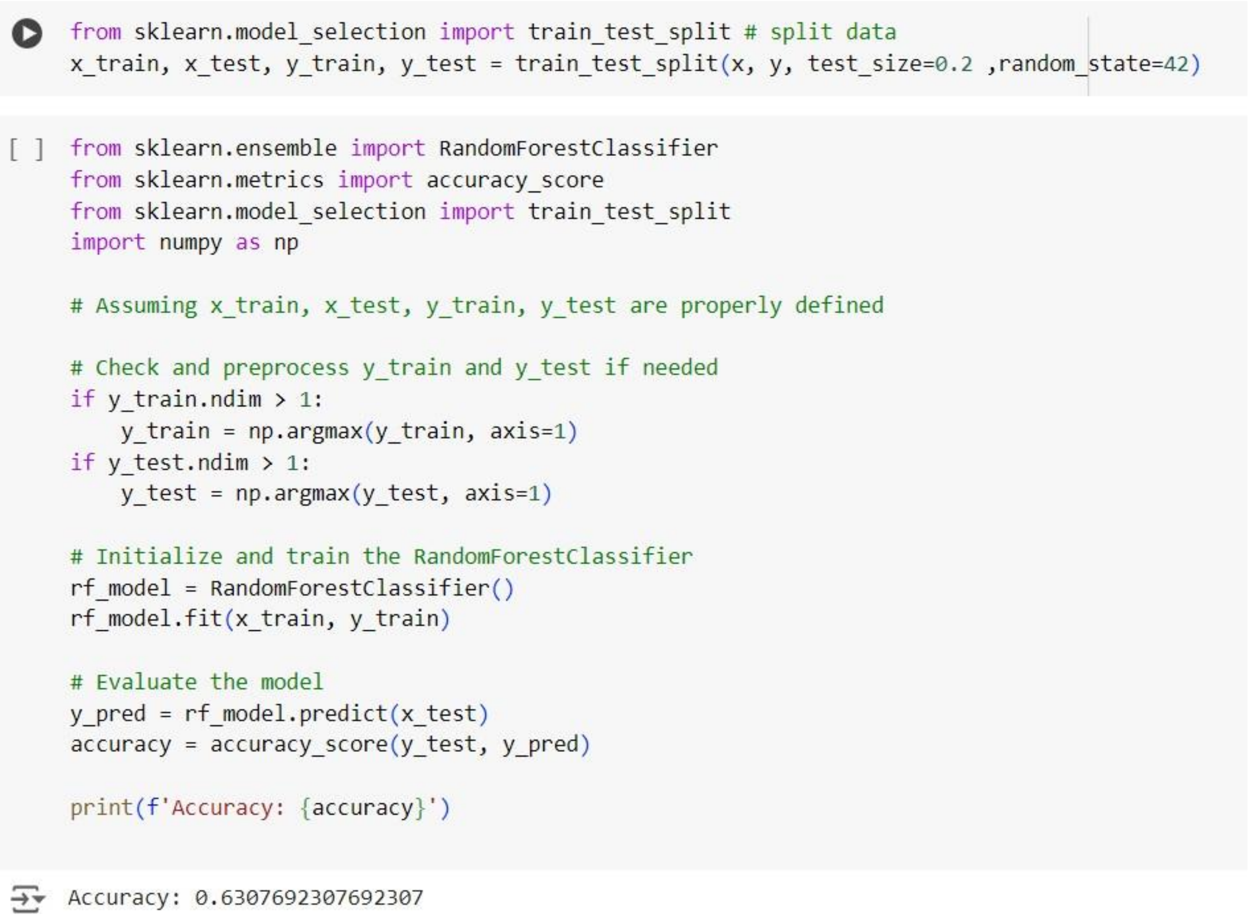
```
from sklearn import preprocessing # labeling  #بتحول الكلامات لارقام عشان تتعامل مع الالجورزم
lb = preprocessing.LabelEncoder()
x.age = lb.fit_transform(x.age)
x.gender = lb.fit_transform(x.gender)
x.Symptom1 = lb.fit_transform(x.Symptom1)
x.Symptom2 = lb.fit_transform(x.Symptom2)
x.Symptom3 = lb.fit_transform(x.Symptom3)
x.Symptom4 = lb.fit_transform(x.Symptom4)
x.Symptom5 = lb.fit_transform(x.Symptom5)
x.Symptom6 = lb.fit_transform(x.Symptom6)
x.Symptom7 = lb.fit_transform(x.Symptom7)
x.Symptom8 = lb.fit_transform(x.Symptom8)
x.Symptom9 = lb.fit_transform(x.Symptom9)
x.Symptom10 = lb.fit_transform(x.Symptom10)
x.Symptom11 = lb.fit_transform(x.Symptom11)
x.Symptom12 = lb.fit_transform(x.Symptom12)
y.diseases = lb.fit_transform(y.diseases)
y.Description = lb.fit_transform(y.Description)

display(y)
```

Figure 15 Convert new data set

4.4.1Building model:-

We use 3 Algorithm:

1-Random forest



```
from sklearn.model_selection import train_test_split # split data
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2 ,random_state=42)
```

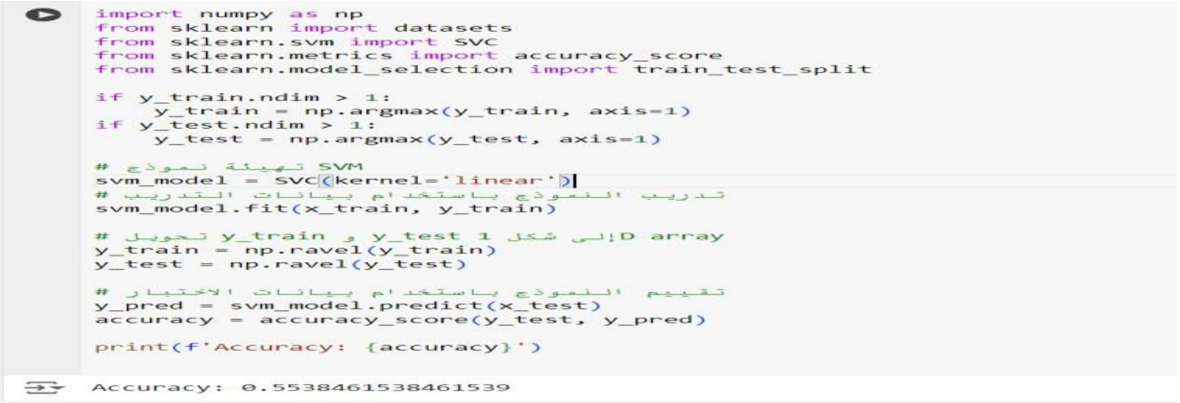
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
import numpy as np

# Assuming x_train, x_test, y_train, y_test are properly defined

# Check and preprocess y_train and y_test if needed
if y_train.ndim > 1:
    y_train = np.argmax(y_train, axis=1)
if y_test.ndim > 1:
    y_test = np.argmax(y_test, axis=1)

# Initialize and train the RandomForestClassifier
rf_model = RandomForestClassifier()
rf_model.fit(x_train, y_train)

# Evaluate the model
y_pred = rf_model.predict(x_test)
accuracy = accuracy_score(y_test, y_pred)

print(f'Accuracy: {accuracy}')
```

```
Accuracy: 0.6307692307692307
```

Figure 1c trains random forest on a new data set

- The result is a bad accuracy of 63%

2-SVM:



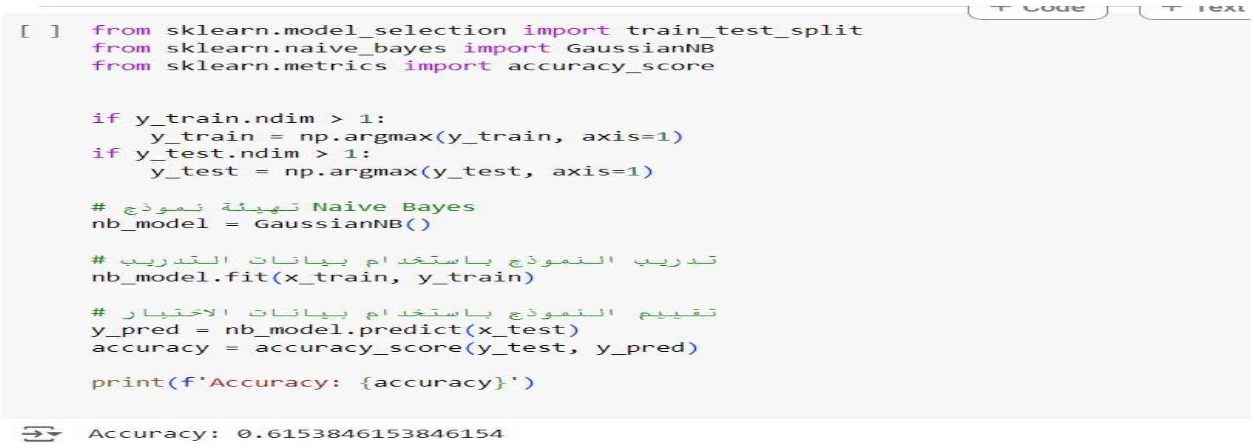
```
import numpy as np
from sklearn import datasets
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

if y_train.ndim > 1:
    y_train = np.argmax(y_train, axis=1)
if y_test.ndim > 1:
    y_test = np.argmax(y_test, axis=1)

# تهيئة نموذج SVM
svm_model = SVC(kernel='linear')
# تدريب النموذج باستخدام بيانات التدريب
svm_model.fit(x_train, y_train)

# تحويل y_train و y_test إلى شكل 1D array
y_train = np.ravel(y_train)
y_test = np.ravel(y_test)

# تقييم النموذج باستخدام بيانات الاختبار
y_pred = svm_model.predict(x_test)
accuracy = accuracy_score(y_test, y_pred)

print(f'Accuracy: {accuracy}')
```

```
Accuracy: 0.5538461538461539
```

Figure 17 trains Svm on a new data set

- The result is a bad accuracy of 55%

3-naïve_base:



```
[ ]  from sklearn.model_selection import train_test_split
     from sklearn.naive_bayes import GaussianNB
     from sklearn.metrics import accuracy_score


     if y_train.ndim > 1:
         y_train = np.argmax(y_train, axis=1)
     if y_test.ndim > 1:
         y_test = np.argmax(y_test, axis=1)

     # تهيئة نموذج Naive Bayes
     nb_model = GaussianNB()

     # تدريب النموذج باستخدام بيانات التدريب
     nb_model.fit(x_train, y_train)

     # تقييم النموذج باستخدام بيانات الاختبار
     y_pred = nb_model.predict(x_test)
     accuracy = accuracy_score(y_test, y_pred)

     print(f'Accuracy: {accuracy}')
```

```
Accuracy: 0.6153846153846154
```

Figure 18 trains naïve based on a new data set

- The result bad accuracy of 61%

- ❖ **The problem: is the accuracy is bad in 3 algorithms.**
- ❖ **The solution: we took refuge in an expert system to solve this problem.**

4.5 Expert system:-

Knowledge Base

1-Loading the Knowledge Base: The Knowledgebase class holds all the rules for diagnosing diseases. Each rule consists of symptoms and corresponding disease information.

2-Populating the Knowledge Base: This part reads the CSV file containing the disease rules and populates the Knowledgebase with these rules. Each row in the CSV file represents a rule with symptoms and corresponding disease details.

Inference Engine

1-Inference Engine Initialization: The Inference Engine class uses the Knowledgebase to infer diseases based on given symptoms. The constructor takes an instance of Knowledgebase.

2-Forward Chaining Algorithm:

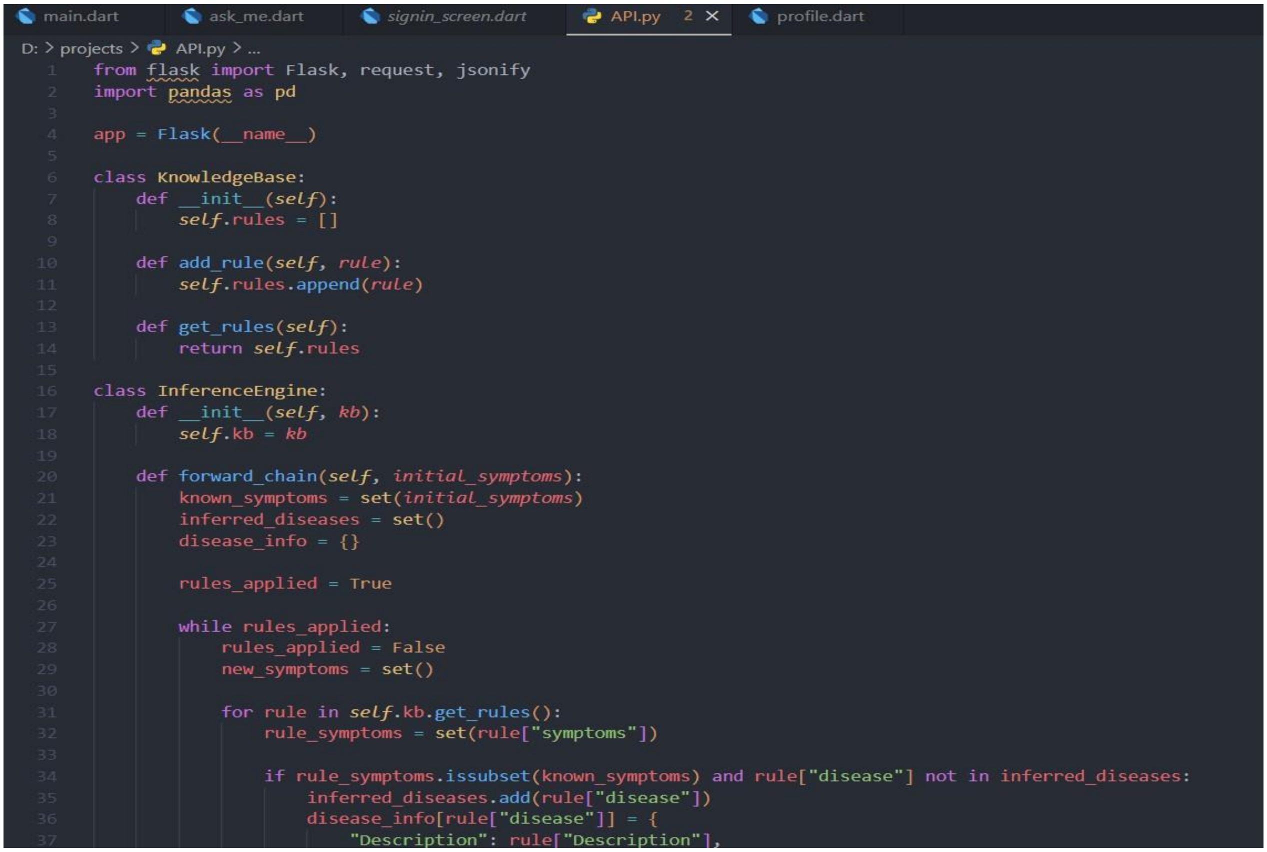
This method implements the forward chaining algorithm:

1-Known Symptoms: A set of symptoms provided by the user.

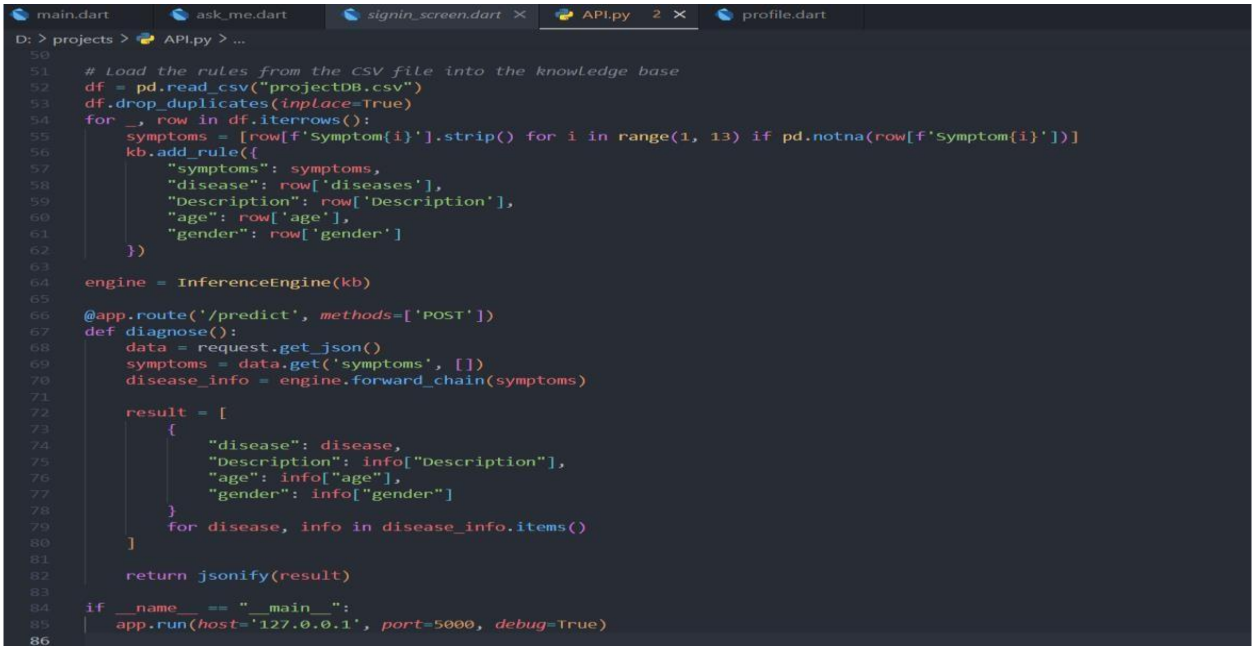
2-Inferred Diseases: A set of diseases inferred based on the known symptoms.

3-Disease Info: A dictionary to store information about the inferred diseases.

The algorithm iterates through the rules in the knowledge base and checks if the symptoms in each rule are a subset of the known symptoms. If a rule's symptoms match, the corresponding disease is inferred, and its information is added to the disease info. The process continues until no more rules can be applied.



```python
from flask import Flask, request, jsonify
import pandas as pd

app = Flask(__name__)

class KnowledgeBase:
    def __init__(self):
        self.rules = []

    def add_rule(self, rule):
        self.rules.append(rule)

    def get_rules(self):
        return self.rules

class InferenceEngine:
    def __init__(self, kb):
        self.kb = kb

    def forward_chain(self, initial_symptoms):
        known_symptoms = set(initial_symptoms)
        inferred_diseases = set()
        disease_info = {}

        rules_applied = True

        while rules_applied:
            rules_applied = False
            new_symptoms = set()

            for rule in self.kb.get_rules():
                rule_symptoms = set(rule["symptoms"])

                if rule_symptoms.issubset(known_symptoms) and rule["disease"] not in inferred_diseases:
                    inferred_diseases.add(rule["disease"])
                    disease_info[rule["disease"]] = {
                        "Description": rule["Description"],
```



```python

# Load the rules from the CSV file into the knowledge base
df = pd.read_csv("projectDB.csv")
df.drop_duplicates(inplace=True)
for _, row in df.iterrows():
    symptoms = [row[f'Symptom{i}'].strip() for i in range(1, 13) if pd.notna(row[f'Symptom{i}'])]
    kb.add_rule({
        "symptoms": symptoms,
        "disease": row['diseases'],
        "Description": row['Description'],
        "age": row['age'],
        "gender": row['gender']
    })

engine = InferenceEngine(kb)

@app.route('/predict', methods=['POST'])
def diagnose():
    data = request.get_json()
    symptoms = data.get('symptoms', [])
    disease_info = engine.forward_chain(symptoms)

    result = [
        {
            "disease": disease,
            "Description": info["Description"],
            "age": info["age"],
            "gender": info["gender"]
        }
        for disease, info in disease_info.items()
    ]

    return jsonify(result)

if __name__ == "__main__":
   app.run(host='127.0.0.1', port=5000, debug=True)

```

Figure 1S Code expert system and flask

4.6 Mobile App Screens:-

1- welcome page:

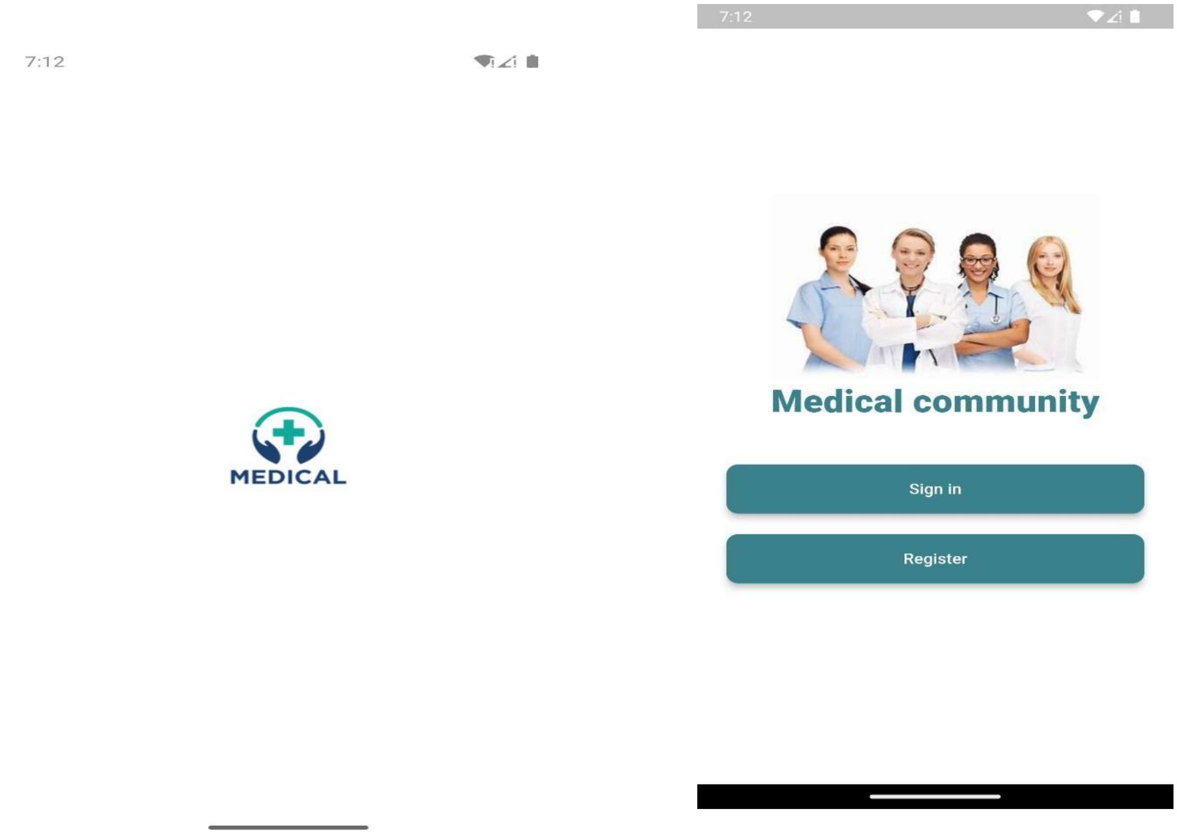


Figure 20 Welcome page

This page contains buttons Sign in and Register, Where the user clicks on sign in if he has previously logged in, or clicks on register if he is logging in for the first time.

2- Sign in page:

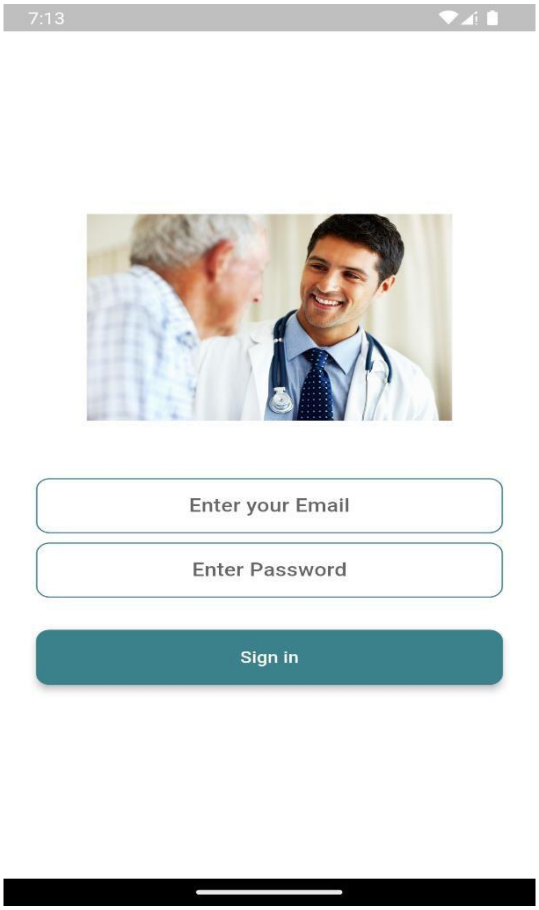


Figure 21 Sign-in page

On the login page, where through that page the user can register with an account to enter the home page so that the application is more secure. This page is linked to the Firebase server. , after entering the email and password, the user's data will go to the Firebase server to verify the account that was registered on the data registration page, and then come to accept the account if it is correct, and if an error occurs, it will bring an error message. There are two cases here. First, if the user does not have an account, he can go to the account registration page, through which he can create a new account. The second case is if the user already has an account and forgot the password, he can go to the forgot password page and enter the account he has, and through it, he can obtain the password.

3- Register page:

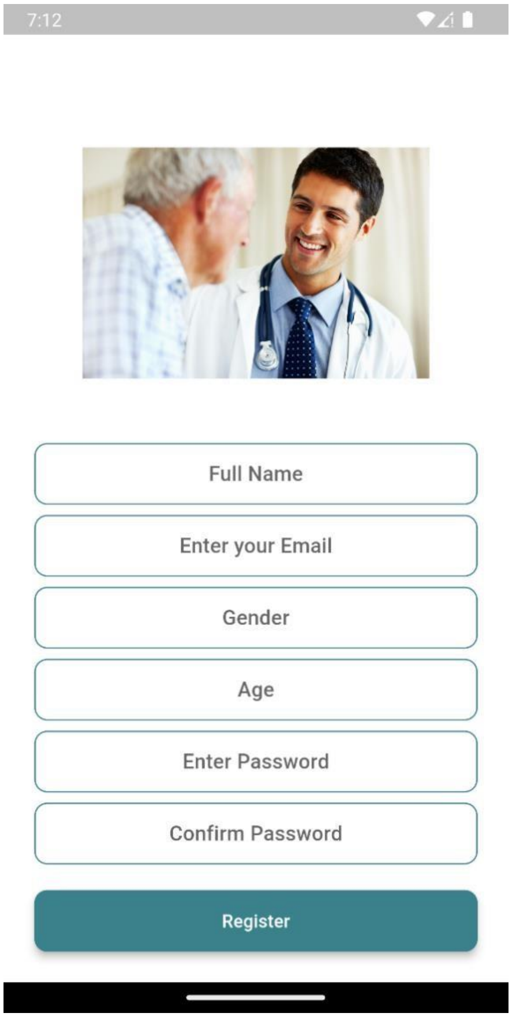
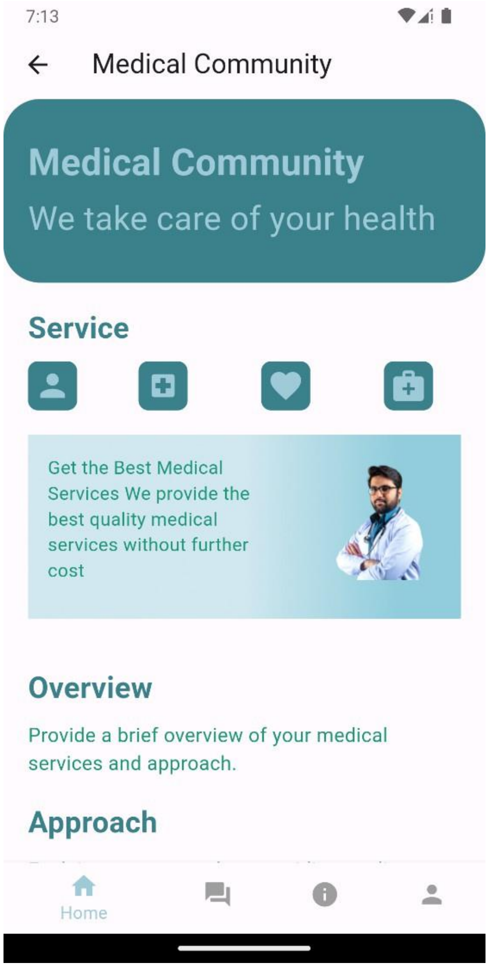


Figure 22 Register page

On the account registration page, the full name, email, gender, age, and password must be entered and confirmed pass, rewritten to send this data to the Firebase server to know that the user wants to create an account so that he can log in to the application and allow him to enter the home page and deal with the application.

4- Home page:



Medical Community

We take care of your health

Get the Best Medical Services We provide the best quality medical services without further cost

Overview

Provide a brief overview of your medical services and approach.

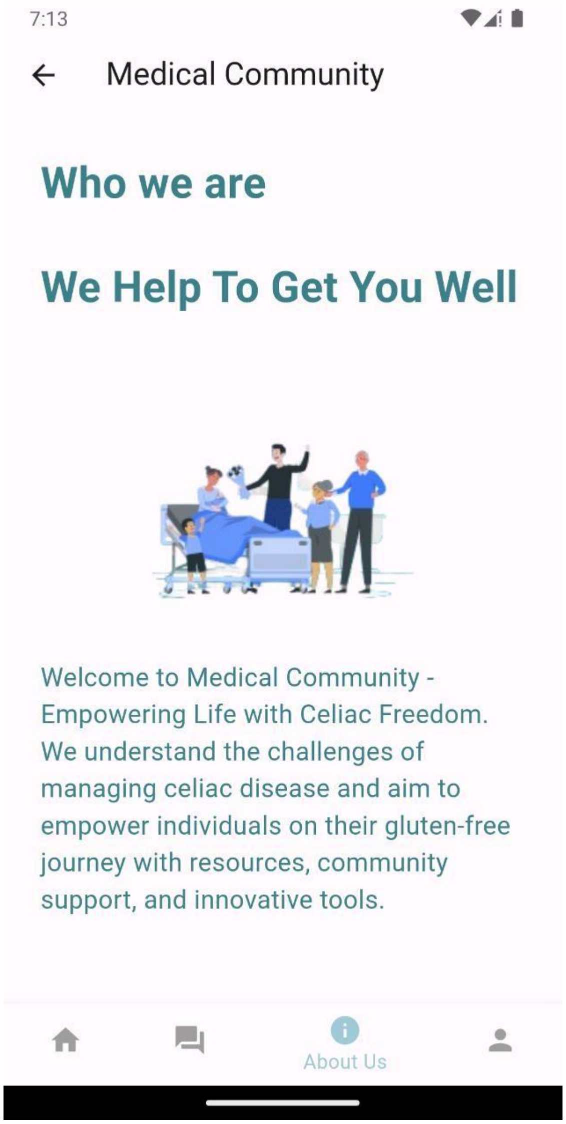
Approach

Figure 23 home page

This page contains service for the patient and overviews us and through it,

You can navigate between pages.

5- About Us page:



7:13

Medical Community

Who we are

We Help To Get You Well

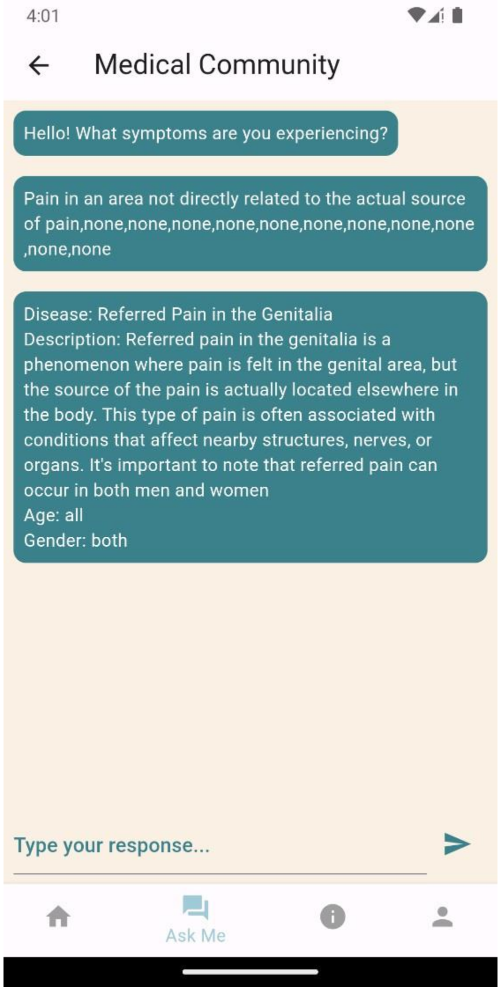
Welcome to Medical Community - Empowering Life with Celiac Freedom. We understand the challenges of managing celiac disease and aim to empower individuals on their gluten-free journey with resources, community support, and innovative tools.

Figure 24 About Us page

This page contains information about us,

And our services and excellence.

6- Ask me page:



Hello! What symptoms are you experiencing?

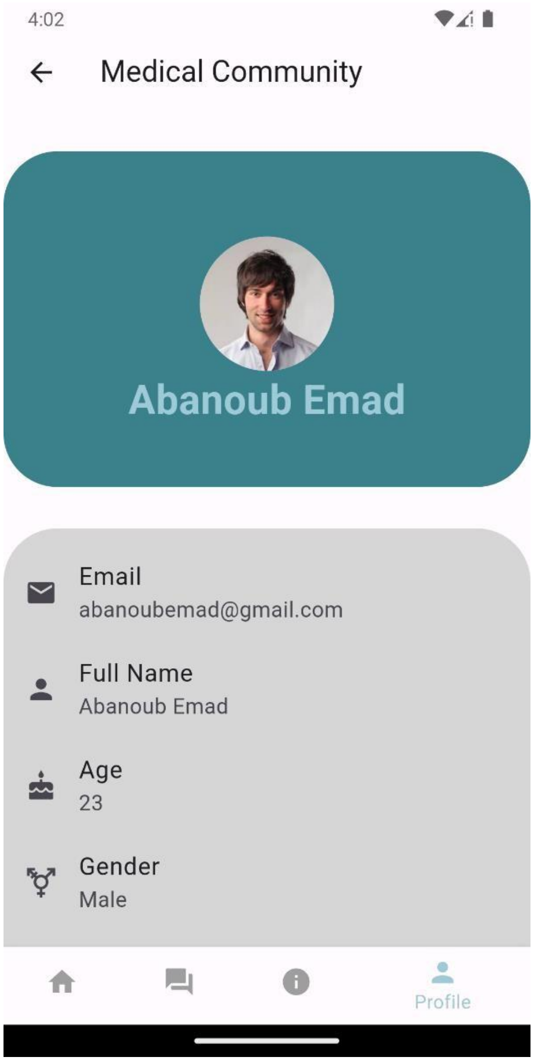
Pain in an area not directly related to the actual source of pain,none,none,none,none,none,none,none,none,none ,none,none

Disease: Referred Pain in the Genitalia
Description: Referred pain in the genitalia is a phenomenon where pain is felt in the genital area, but the source of the pain is actually located elsewhere in the body. This type of pain is often associated with conditions that affect nearby structures, nerves, or organs. It's important to note that referred pain can occur in both men and women
Age: all
Gender: both

Figure 25 Ask me page

This page is a chatbot through which the patient can express the symptoms he feels. He enters the symptoms he feels at the bottom of the screen, then presses enter, and then the chat responds to him with the disease he is afflicted with and a description of the disease, age, and gender, which is a rudimentary diagnosis for the patient.

7- Profile page:



Abanoub Emad

Email
abanoubemad@gmail.com

Full Name
Abanoub Emad

Age
23

Gender
Male

Figure 2c profile page

This page stores the information of the user who registered on the login page, which contains the email, full name, gender, and age.

Chapter 5

Tools And Technologies

5.1 Technologies:-

5.1.1 python:

What is Python?

Python is a high-level, general-purpose, interpreted object-oriented programming language. Similar to PERL, Python is a programming language popular among experienced C++ and Java programmers.

Working in Python, users can interpret statements in several operating systems, including UNIX-based systems, Mac OS, MS-DOS, OS/2, and various versions of Microsoft Windows 10 and Windows 11.

Python's origins and benefits:-

Python emerged three decades ago. Its inventor, Dutch programmer, Guido van Rossum, named it after his favorite comedy group at the time, Monty Python's Flying Circus. Since then,

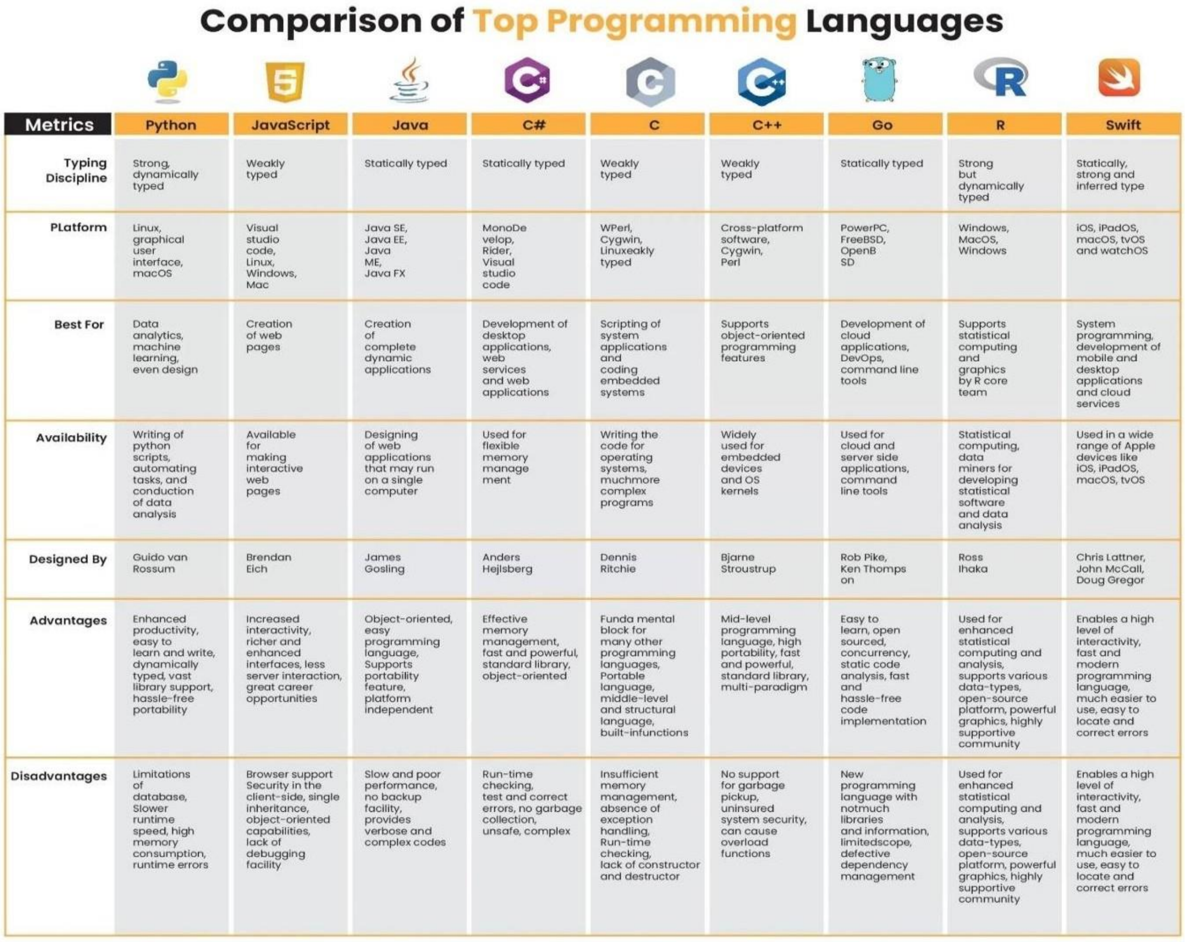
it has attracted a vibrant community of enthusiasts who work on fixing potential bugs and extending the capabilities of the code.

Python is known for being powerful, and fast and for making programming more fun. Python coders can dynamically type variables without having to explain what the variable is supposed to be. Users can download Python at no cost and start learning to code with it right away. The source code is freely available and open for modification and reuse.

Python adoption is widespread because of its clear syntax and readability. Used often in data analytics, machine learning (ML), and web development, Python yields code that is easy to read, understand and learn. Python's indentation requirements for source statements help make the code consistent and easy to read. Applications developed with Python code tend to be smaller than software built with programming languages like Java. Programmers generally have to type less code.

Python programming also remains popular because the interpreter is excellent at discovering bugs and raising an exception. In this case, bad inputs never trigger a segmentation fault. As the debugger is Python-based, users won't have to worry about any potential conflicts.

Python continues to grow and is actively used by some of the largest multinationals and corporations that also support Python with guides, tutorials, and resources.



Comparison of Top Programming Languages

Metrics	Python	JavaScript	Java	C#	C	C++	Go	R	Swift
Typing Discipline	Strong, dynamically typed	Weakly typed	Statically typed	Statically typed	Weakly typed	Weakly typed	Statically typed	Strong but dynamically typed	Statically, strong and inferred type
PLatform	Linux, graphical user interface, macOS	Visual studio code, Linux, Windows, Mac	Java SE, Java EE, Java ME, Java FX	MonoDe velop, Rider, Visual studio code	WPerl, Cygwin, Linuxeakly typed	Cross-platform software, Cygwin, Perl	PowerPC, FreeBSD, OpenB SD	Windows, MacOS, Windows	iOS, iPadOS, macOS, tvOS and watchOS
Best For	Data analytics, machine learning, even design	Creation of web pages	Creation of complete dynamic applications	Development of desktop applications, web services and web applications	Scripting of system applications and coding embedded systems	Supports object-oriented programming features	Development of cloud applications, DevOps, command line tools	Supports statistical computing and graphics by R core team	System programming, development of mobile and desktop applications and cloud services
Availability	Writing of python scripts, automating tasks, and conduction of data analysis	Available for making interactive web pages	Designing of web applications that may run on a single computer	Used for flexible memory manage ment	Writing the code for operating systems, muchmore complex programs	Widely used for embedded devices and OS kernels	Used for cloud and server side applications, command line tools	Statistical computing, data miners for developing statistical software and data analysis	Used in a wide range of Apple devices like iOS, iPadOS, macOS, tvOS
Designed By	Guido van Rossum	Brendan Eich	James Gosling	Anders Hejlsberg	Dennis Ritchie	Bjarne Stroustrup	Rob Pike, Ken Thomps on	Ross Ihaka	Chris Lattner, John McCall, Doug Gregor
Advantages	Enhanced productivity, easy to learn and write, dynamically typed, vast library support, hassle-free portability	Increased interactivity, richer and enhanced interfaces, less server interaction, great career opportunities	Object-oriented, easy programming language, Supports portability feature, platform independent	Effective memory management, fast and powerful, standard library, object-oriented	Funda mental block for many other programming languages, Portable language, middle-level and structural language, built-infunctions	Mid-level programming language, high portability, fast and powerful, standard library, multi-paradigm	Easy to learn, open sourced, concurrency, static code analysis, fast and hassle-free code implementation	Used for enhanced statistical computing and analysis, supports various data-types, open-source platform, powerful graphics, highly supportive community	Enables a high level of interactivity, fast and modern programming language, much easier to use, easy to locate and correct errors
Disadvantages	Limitations of database, Slower runtime speed, high memory consumption, runtime errors	Browser support Security in the client-side, single inheritance, object-oriented capabilities, lack of debugging facility	Slow and poor performance, no backup facility, provides verbose and complex codes	Run-time checking, test and correct errors, no garbage collection, unsafe, complex	Insufficient memory management, absence of exception handling, Run-time checking, lack of constructor and destructor	No support for garbage pickup, uninsured system security, can cause overload functions	New programming language with notmuch libraries and information, limitedscope, defective dependency management	Used for enhanced statistical computing and analysis, supports various data-types, open-source platform, powerful graphics, highly supportive community	Enables a high level of interactivity, fast and modern programming language, much easier to use, easy to locate and correct errors

Figure 27 Programming languages

Python use cases:-

Python offers dynamic data types, ready-made classes, and interfaces to many system calls and libraries. Users can also extend it using another programming language like C or C++. Its high-level data structures, dynamic binding, and dynamic typing make it one of the go-to programming languages for rapid application development.

Python also is often used as a glue or scripting language that seamlessly connects existing components. Users can use it for scripting in Microsoft's Active Server Page technology.

Primary use cases for Python include the following:

ML

server-side web development

software development

system scripting

Anyone who uses Facebook, Google, Instagram, Reddit, Spotify, or YouTube has encountered Python code. Python code can also be found in the scoreboard system for the Melbourne (Australia) Cricket Ground. Z Object Publishing Environment, a popular web Application Server, is written in Python.

Python training and tools:-

As a result of extensive community support and a syntax that stresses readability, Python is relatively easy to learn. Some online courses offer to teach users Python programming in six weeks.

Python itself also provides modules and packages to learn and supports program modularity and code reuse. As users work with Python, they will want to be familiar with the current version, development environment, and supporting tools, specifically the following:

Python 3.0, which dates to 2008, remains the latest version. Unlike previous updates that concentrated on debugging earlier versions of Python, Python 3 had forward compatibility and coding style changes. As a result, Python 3 could not support previous releases. The code syntax narrowed in on code repetition and redundancy, allowing the code to tackle the same tasks in many different ways. This single change made it much easier for beginners to learn Python programming.

Integrated Development and Learning Environment (IDLE) is the standard Python development environment. It enables access to the Python interactive mode through the Python shell window. Users can also use Python IDLE to create or edit existing Python source files by leveraging the file editor.

Python Launcher lets developers run Python scripts from the desktop. Simply select Python Launcher as the default application to open any .py script by double-clicking on it through the Finder window. Python Launcher offers many options to control how users launch Python scripts.

Anaconda is a leading open-source distribution for Python and R programming languages with over 300 built-in libraries specially developed for ML projects. Its primary objective is to simplify package management and deployment.

5.1.2 Flutter:-

Flutter is an open-source UI software development kit created by Google. It can be used to develop cross-platform applications from a single codebase for the web, Fuchsia, Android, iOS, Linux, macOS, and Windows. First described in 2015, Flutter was released in May 2017. Flutter is used internally by Google in apps such as Google Pay and Google Earth as well as by other software developers including Byte Dance and Alibaba.

Flutter consists of both a UI language and a rendering engine. When a Flutter application is compiled, it ships with both the UI code and the rendering engine, which is about 4 MB compressed This is in contrast to many other UI frameworks that rely on a separate rendering engine and only ship the UI code, such as native Android apps which rely on the device-level Android SDK or HTML/JavaScript Web apps that rely on the user's HTML engine and JavaScript engine. Flutter's complete control of its rendering pipeline makes supporting multiple platforms simpler as it only needs the platform to support running native code such as via the Android Java Native Interface rather than support Flutter's UI model in its entirety.

In the last few years of this decade, we have seen a lot of app startups emerging from all across the globe. With the rise in technology and the availability of smartphones, many startups find it easy to connect with users and clients via apps. The app market has also grown in the last few years and is expected to grow exponentially in the coming decade. The app development market has also been on the rise and has allowed countless app developers to exhibit their skills and find suitable jobs. With this shift into apps, much development, and research have been done to deliver the best and to make the app development process faster and much simpler. Apps can be broadly categorized as:

1. iOS Apps:

These apps are made for Apple devices and wear. iOS apps are made using the Swift language. The iOS apps have an extension of .ipa.

2. Android Apps:

These apps are made for Android devices and wear. Android apps are made using Java and Kotlin, with an extension of .apk. Many app developers who had to work in a cross-platform work environment, and are responsible for the development of both Android and iOS apps, found it a difficult and lengthy process to develop apps for both platforms. The major problems encountered by companies and developers were:

No Cross-Platform Dependency: iOS and Android apps work very differently internally, so the developers had to redesign and reconfigure the same content for individual platforms.

Time Constraints: Making a professional app, from coding to designing, requires a lot of time. Companies usually set a time limit by which the app should be ready to be launched into the market. Those developers who had to work on both these platforms often found time limit issues, and the efficiency and quality of work degraded.

More Employees: This problem was encountered by companies. Since they have to develop an app for both platforms, a greater number of app developers knowing about the individual platform had to be hired.

Development Cost: Since the app has to be made individually for both platforms, the cost of development will increase, as more developers will be required.

Since the launch of Flutter in May 2017, it has resolved many of the existing problems in the app development industry. Flutter is a powerful technology, or we can say a tool backed by Dart language packed with a powerful mobile framework that can be used in both iOS and Android applications. Flutter is often used with DART, which is an object-oriented programming language by Google. The Flutter development tools come with a graphics library and material design, and the Cupertino design allows faster operations of the app and also gives the app a stunning look, irrespective of its operating platform! The biggest advantage of Flutter is that it can be used to create cross-platform apps. Using Flutter, one can create iOS apps, Android apps, Websites, and many cross-platform software in just one go, there is no need to write code for different platforms.

Features of flutter:-

Flutter structure offers the accompanying elements to designers:

Present day and receptive structure.

Utilizes Dart programming language, and it is extremely simple to learn.

A quick turn of events.

Delightful and liquid UIs.

Colossal gadget list.

Runs the same UI for numerous stages.

Superior execution application

Fast and responsive layout.

Easy connection of back-end and synchronization.

Advantages:-

1. Cross-platform Operations: Apps made with Flutter can be operated on both platforms (iOS and Android). There is no need for reconfiguration and redesigning.

2. Less Need for Developers: This can be advantageous for the companies, as they require a smaller number of developers and the app can also work on both platforms.

3. Less Development Cost: Since there are a smaller number of developers needed, the cost incurred for the development of the app also reduces.

4. Time Constraint: The time required to launch the app into the market, also reduces as only a single app has to be made, which would work independently of the platform.

5. Powerful Design: The Flutter mobile framework is the latest in the market, and this helps to create a very powerful app design with minimum effort.

Disadvantages of flutter:-

Despite its many benefits, vacillation has the accompanying downsides:

1- However, since it is coded in Dart language, a designer needs to learn a new dialect, although it is not difficult to learn.

2- The current system attempts to isolate rationale and UI however much as could be expected at the same time, in Shudder,

UI and rationale are intermixed. We can beat this by utilizing savvy coding and

utilizing significant-level modules to isolate UI and rationale.

3-Ripple is one more system to make versatile applications. Designers are having

a tough time in picking the right improvement devices in an immensely populated portion

5.1.3 Firebase:-

What is Firebase?

Firebase is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google’s infrastructure.

Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents.

Key Features

1. Authentication

It supports authentication using passwords, phone numbers, Google, Facebook, Twitter, and more. The Firebase Authentication (SDK) can be used to manually integrate one or more sign-in methods into an app.

2. Realtime database

Data is synced across all clients in real-time and remains available even when an app goes offline.

3. Hosting

Firebase Hosting provides fast hosting for a web app; content is cached into content delivery networks worldwide.

4. Test lab

The application is tested on virtual and physical devices located in Google’s data centers.

5. Notifications

Notifications can be sent with Firebase with no additional coding.

Users can get started with Firebase for free; more details can be found on the official

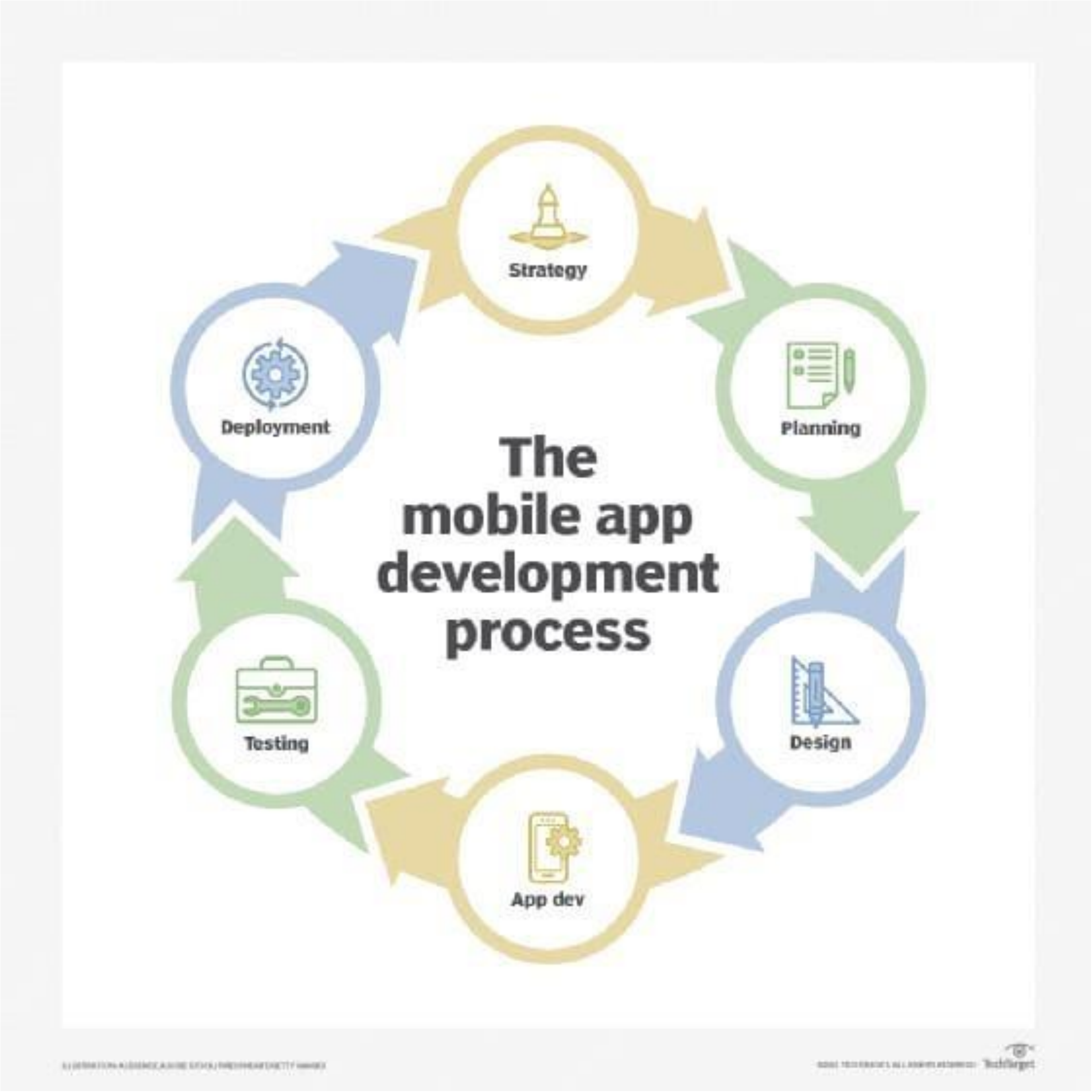


Figure 28 mobile process

What is Google Firebase secure?

All of Firebase's data centers are SOC 2 Type 2 and ISO 27001 certified. Firebase also uses a variety of security measures to protect your data, including the following:

Data encryption. All of Firebase's data is encrypted at rest and in transit.

Role-based access control. Firebase uses role-based access control (RBAC) to give granular control over who can access application data.

Audit logging. Firebase logs all access to data, so businesses can track who has accessed application data and when.

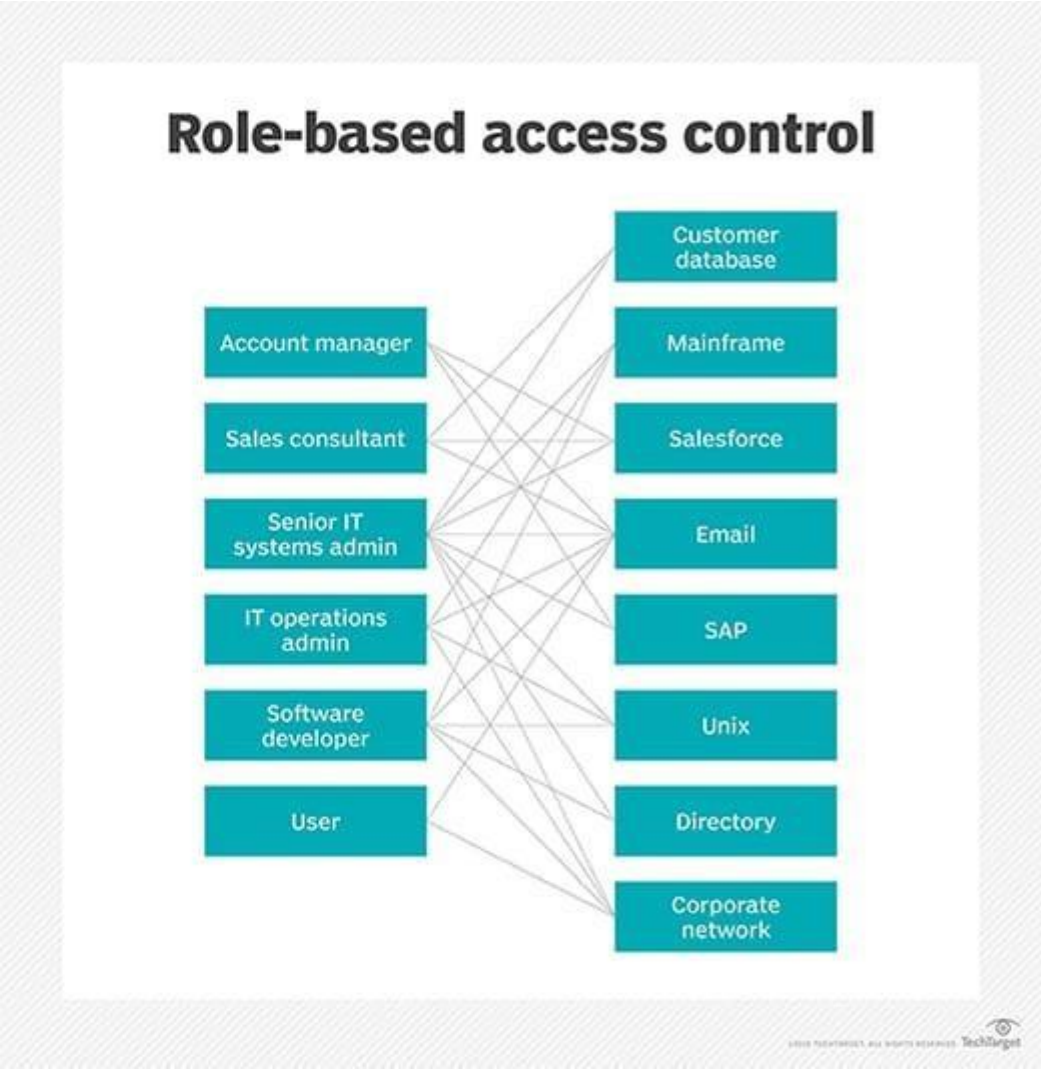


Figure 2S access control

5.1.4 API (Flask):-

What is Flask?

Python’s Flask micro web framework can be used to build web-based apps for interacting and displaying data-driven content. Although Flask is more frequently associated with web development, it can also be used in data science to create straightforward web interfaces, APIs (Application Programming Interfaces), and visualization tools that enable data scientists and analysts to present their data and analyses in a user-friendly and interactive way.

Flask: Ideal for smaller projects or prototypes due to its lightweight nature and minimalistic design. Offers flexibility and simplicity, making it suitable for rapid development.

Flasks can be used for making solutions or for holding, containing, collecting, or sometimes volumetrically measuring chemicals, samples, solutions, etc. for chemical reactions or other processes such as mixing, heating, cooling, dissolving, precipitation, boiling (as in distillation), or analysis.

Flask is a small and lightweight Python web framework that provides useful tools and features that make creating web applications in Python easier. It gives developers flexibility and is a more accessible framework for new developers since you can build a web application quickly using only a single Python file. Flask is also extensible and doesn’t force a particular directory structure or require complicated boilerplate code before getting started.

As part of this tutorial, you’ll use the Bootstrap toolkit to style your application so it is more visually appealing. Bootstrap will help you incorporate responsive web pages in your web application so that it also works well on mobile browsers without writing your own HTML, CSS, and JavaScript code to achieve these goals. The toolkit will allow you to focus on learning how Flask works.

Flask uses the Jinja template engine to dynamically build HTML pages using familiar Python concepts such as variables, loops, lists, and so on. You'll use these templates as part of this project.

In this tutorial, you'll build a small web blog using Flask and SQLite in Python 3. Users of the application can view all the posts in your database and click on the title of a post to view its contents with the ability to add a new post to the database and edit or delete an existing post.

Common Uses of Flask:-

- Web Applications: Flask builds various types of web applications, including blogs, e-commerce sites, social media platforms, and more. Its simplicity and flexibility make it suitable for both small projects and larger, more complex applications.
- API Development: Create RESTful APIs that enable communication between different software systems. Use these APIs to share data, perform actions, and integrate various services.
- Prototyping: Flask's lightweight nature makes it a great choice for quickly prototyping web-based ideas and concepts. It allows developers to rapidly create and test their ideas without the overhead of more complex frameworks.
- Microservices: Flask lends itself well to constructing microservices, which constitute small, autonomously deployable constituents within more extensive applications. Developers can construct each microservice utilizing Flask to offer distinct functionalities.
- Webhooks: Webhooks function by obtaining data from external services upon the occurrence of particular events. Flask enables the creation of endpoints capable of listening to these events and initiating actions within your application.
- Interactive Dashboards: Flask can be used to build interactive dashboards that visualize data in real time. This is useful for data analysis, reporting, and monitoring purposes.
- Educational Projects: Flask is often used in educational settings to teach web development concepts due to its simplicity and clear structure. Students can

quickly learn about routes, templates, and interactions between the front end and back end.

- Small to Medium-sized Websites: For websites that don’t require the complexity of larger frameworks, Flask provides a lightweight alternative that still supports various features.
- Integration with Data Science and Machine Learning: Flask empowers the creation of web interfaces for data science and machine learning models, facilitating user interaction and utilization of these models without the necessity of comprehending the underlying code.

What is a Framework?

The framework is the basis upon which software programs are built. It serves as a foundation for software developers, allowing them to create a variety of applications for certain platforms. A set of functions and predefined classes is used to connect with the system software and handle inputs and outputs. The life of a developer gives them the ability to use certain extensions and makes the online applications scalable and maintainable.

What are the key features of Flask?

Flask is a popular Python web framework that allows you to create web applications quickly and easily. Some of the key features of Flask are:

- Lightweight and minimalistic: Flask has very few dependencies and provides only the essential components for web development, such as routing, request handling, templating, and testing. This makes Flask easy to learn and use and also gives you more flexibility and control over your application.
- Jinja2 templating engine: Flask uses Jinja2 as its default templating engine, which lets you write dynamic HTML templates with Python-like syntax. Jinja2 supports features such as inheritance, macros, filters, and expressions, and also offers security features such as auto-escaping and sandboxing.

- WSGI compliant: Flask is compliant with the Web Server Gateway Interface (WSGI) standard, which defines how web servers and web applications communicate. This means that Flask can run on any WSGI-compatible web server, such as Guni corn, wigs, or Apache.
- Extensible and modular: Flask supports a modular approach to web development, where you can organize your application into smaller and reusable components called blueprints. Flask also has a rich ecosystem of extensions that provide additional functionality, such as database integration, authentication, caching, email, and more. You can choose the extensions that suit your needs and customize them as you like.

5.2 Libraries:-

5.2.1 pandas:

What is Pandas Library in Python?

Pandas is a powerful and versatile library that simplifies the tasks of data manipulation in Python.

Pandas are well-suited for working with tabular data, such as spreadsheets or SQL tables.

The Pandas library is an essential tool for data analysts, scientists, and engineers working with structured data in Python.

Pandas is a Python library used for working with data sets.

It has functions for analyzing, cleaning, exploring, and manipulating data.

The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.

What is Python Pandas used for?

The Pandas library is generally used for data science, but have you wondered why? This is because the Pandas library is used in conjunction with other libraries that are used for data science.

It is built on top of the NumPy library which means that a lot of the structures of NumPy are used or replicated in Pandas.

The data produced by Pandas is often used as input for plotting functions in Matplotlib, statistical analysis in SciPy, and machine learning algorithms in Scikit-learn.

You must be wondering, Why should you use the Pandas Library. Python’s Pandas library is the best tool to analyze, clean, and manipulate data.

Here is a list of things that we can do using Pandas.

Data set cleaning, merging, and joining.

Easy handling of missing data (represented as Nan) in floating point as well as non-floating point data.

Columns can be inserted and deleted from the Data Frame and higher-dimensional objects.

Powerful group by functionality for performing split-apply-combine operations on data sets.

Data Visualization.

Pandas give you answers about the data. Like:

Is there a correlation between two or more columns?

What is the average value?

Max value?

Min value?

Pandas are also able to delete rows that are not relevant or contain wrong values, like empty or NULL values. This is called cleaning the data.

Pandas allow us to analyze big data and make conclusions based on statistical theories.

Pandas can clean messy data sets, and make them readable and relevant.

How Pandas Works:-

Included in the Pandas open-source library are Data Frames, which are two-dimensional array-like data tables in which each column contains values of one variable and each row contains one set of values from each column. Data stored in a Data Frame can be of numeric, factor, or character types. Pandas Data Frames are also thought of as a dictionary or collection of series objects.

Data scientists and programmers familiar with the R programming language for statistical computing know that Data Frames are a way of storing data in grids that are easily overviewed. This means that Pandas are chiefly used for machine learning in the form of Data Frames.

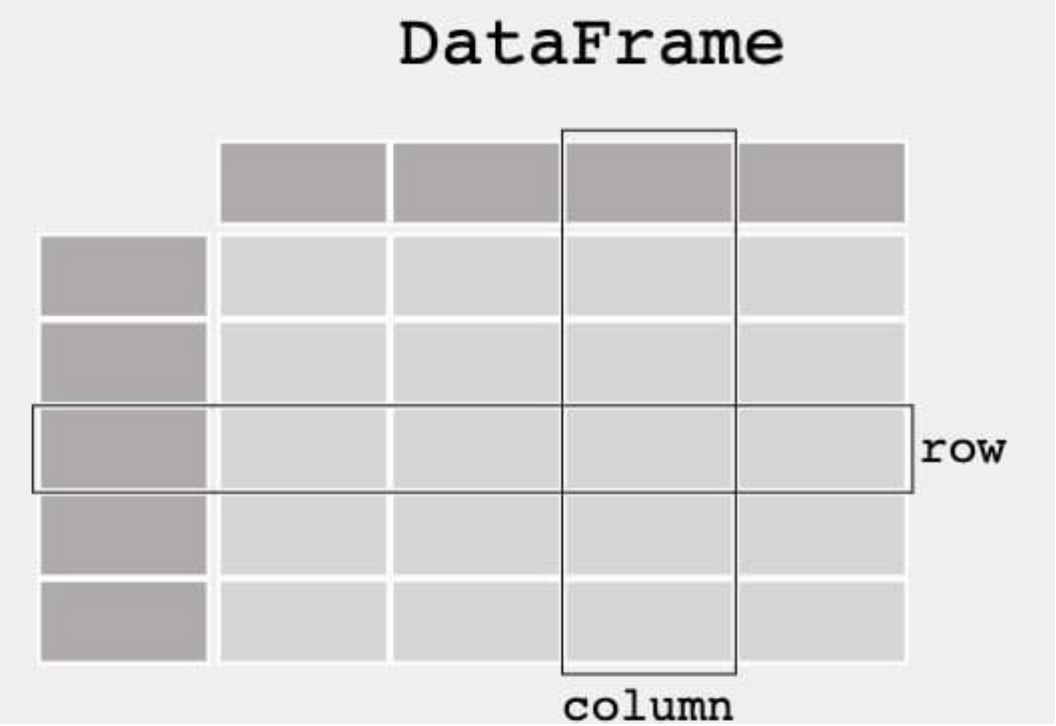


Figure 30 How pandas work

Pandas allow for importing and exporting tabular data in various formats, such as CSV or JSON files.

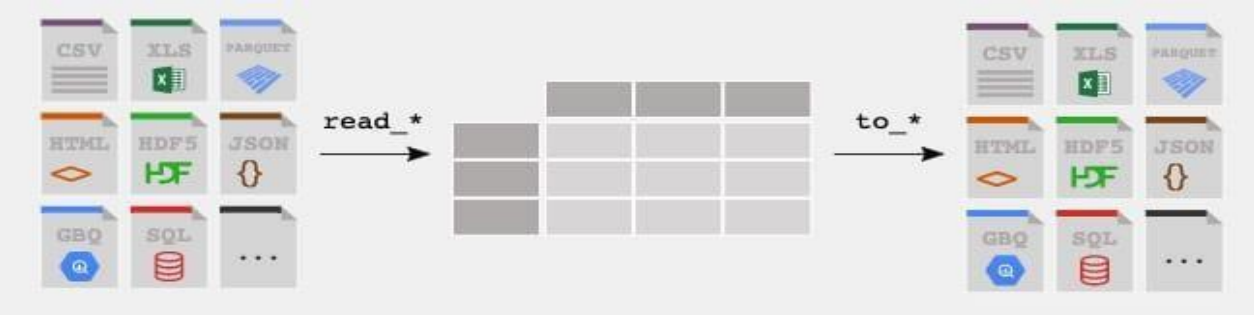


Figure 31 How pandas work

Pandas also allows for various data manipulation operations and data cleaning features, including selecting a subset, creating derived columns, sorting, joining, filling, replacing, summary statistics, and plotting.

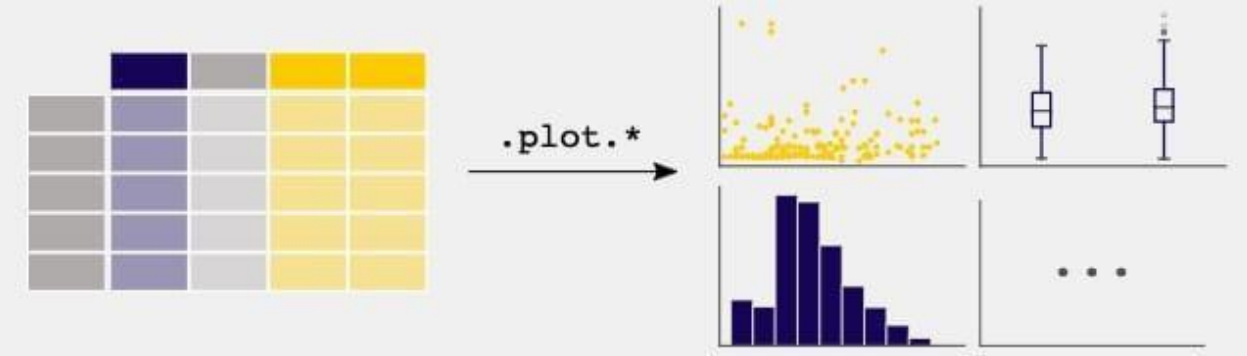


Figure 32 How pandas work

Benefits of Pandas:-

Again according to the Python Package Index organizers, Pandas delivers several key benefits to data scientists and developers alike, including:

1- Easy handling of missing data (represented as Nan) in both floating point and non-floating point data

2- Size mutability: columns can be inserted and deleted from Data Frames and higher-dimensional objects

3- Automatic and explicit data alignment: objects can be explicitly aligned to a set of labels, or the user can simply ignore the labels and let series,

Data Frame, etc. automatically aligns the data in computations

4- Powerful, flexible group-by functionality to perform split-apply-combine operations on data sets for both aggregating and transforming data

5- Making it easy to convert ragged, differently indexed data in other Python and NumPy data structures into Data Frame objects

6- Intelligent label-based slicing, fancy indexing, and sub-setting of large data sets

7- Intuitive merging and joining of data sets

8- Flexible reshaping and pivoting of data sets

9- Hierarchical labeling of axes (possible to have multiple labels per tick)

10- Robust I/O tools for loading data from flat files (CSV and delimited), Excel files, databases, and saving/loading data from the ultrafast HDF5 format

11-Time series-specific functionality: date range generation and frequency conversion, moving window statistics, date shifting, and lagging

5.2.2 Firebase authentication:-

What is Firebase authentication:

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, and popular federated identity providers like Google, Facebook Twitter, and more.

Firebase Authentication integrates tightly with other Firebase services, and it leverages industry standards like OAuth 2.0 and OpenID Connect, so it can be easily integrated with your custom backend.

Firebase UI is a library provided by Firebase for Android apps which makes so many tasks easy while integrating Firebase in Android. This library provides so many extra features that we can integrate into our Android very easily. In this article, we will take a look at using this library for adding authentication in our Android apps.

Firebase Authentication is a crucial part of most applications. It provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. This article will provide an overview of Firebase Authentication, its features, benefits, and the platforms it supports.

Overview of Firebase Authentication:

Firebase Authentication is a service provided by Firebase, a platform developed by Google for creating mobile and web applications. It offers a complete end-to-end identity solution, supporting email and password accounts, phone auth, Google, Twitter, Facebook, and GitHub login, and more. Firebase Authentication is designed to be easy to use, providing developers with a simple way to authenticate their users while also ensuring their applications remain secure.

Firebase Authentication Features and Benefits:

Firebase Authentication comes with a host of features that make it a popular choice for developers. Some of these features include:

1- **Multiple Authentication Providers**: Firebase Authentication supports a variety of sign-in methods, including email/password, phone, Google, Facebook, Twitter, and GitHub.

2- **Secure and Reliable**: Firebase Authentication uses industry-standard protocols to ensure the security of user data. It also provides automatic detection of suspicious activity and can block or warn about suspicious users.

3- **Scalable**: Firebase Authentication can handle large numbers of users without any decrease in performance, making it suitable for applications of all sizes.

3- **Customizable**: Developers can customize the user experience, including the sign-in and sign-up flows, to match the look and feel of their application.

4- **Integration with Other Firebase Services**: Firebase Authentication integrates seamlessly with other Firebase services, such as Firestore and Cloud Functions, allowing developers to create secure, feature-rich applications.

Understanding the Importance of User Authentication:

User authentication is a fundamental part of any application that restricts access to certain areas or features. It verifies the identity of users, ensuring that each user is who they claim to be. This process is essential for protecting sensitive information and providing a personalized user experience.

Supported Platforms for Firebase Authentication

Firebase Authentication supports a wide range of platforms, including:

- iOS: Firebase provides an SDK with both Objective-C and Swift interfaces for integrating with iOS applications.

- Android: Firebase provides an SDK for integrating with Android applications.
- Web: Firebase provides a JavaScript SDK for integrating with web applications. It supports both mobile and desktop web.
- Unity: Firebase provides a Unity SDK for integrating with Unity applications.
- C++: Firebase provides a C++ SDK, primarily aimed at game developers.

In conclusion, Firebase Authentication is a powerful, flexible, and secure solution for user authentication. It provides a host of features that make it easy for developers to authenticate their users, protect sensitive data, and create personalized user experiences. Whether you're developing a small personal project or a large-scale commercial application, Firebase Authentication has the tools you need to ensure your users' identities are verified and protected.

5.2.3 Firebase Core:-

Firebase Core is a fundamental part of the Firebase platform, providing the essential services needed to integrate other Firebase features into your app. Here are some key uses and features of Firebase Core:

1- Configuration Management:

- Initialization: Firebase Core handles the initial setup and configuration of your Firebase app, making it easier to integrate other Firebase services.
- Google Services Configuration: It uses the google-services. Jason (for Android) or Google Service-Info. List (for iOS) file to automatically configure the app with the necessary Firebase project settings.

2- Analytics:

- Google Analytics: Firebase Core often integrates with Firebase Analytics, providing insights into user behavior and app performance. This helps in making data-driven decisions to improve user engagement and retention.

3- App Monitoring:

- Crashlytics: It integrates with Firebase Crashlytics to provide real-time crash reports and diagnostics, helping developers fix issues quickly.
- Performance Monitoring: Firebase Core works with Performance Monitoring to track the performance of your app, including network requests, app startup time, and other custom metrics.

4- User Authentication:

- Authentication: Firebase Core can integrate with Firebase Authentication, making it easier to manage user authentication and identity services.

5- Cloud Messaging:

- Notifications: It works with Firebase Cloud Messaging (FCM) to enable push notifications, allowing developers to send targeted messages and notifications to users.

6- Remote Config:

- Configuration Updates: Firebase Core helps in using Firebase Remote Config to change the behavior and appearance of your app without requiring users to download an app update.

7- Cloud Storage and Fire store:

- Data Management: It provides the groundwork for integrating Firebase Fire Store (a NoSQL database) and Firebase Cloud Storage, simplifying data storage and retrieval.

8- Ad Mob Integration:

- Monetization: Firebase Core supports Ad Mob integration, enabling ad monetization and tracking ad performance.

9- Dynamic Links:

- Deep Linking: It facilitates the use of Firebase Dynamic Links for creating deep links that work across different platforms and installation states.

10- App Distribution:

- Beta Testing: Firebase Core supports Firebase App Distribution, allowing developers to distribute pre-release versions of their app to testers.

5.2.4 Cloud Fire Store:-

What is the Google Fire store?

powerful database is quite important for any business that intends to host high-performance applications for mobile and web platforms. Easy access to well-organized information, effortless data management, and robust automation features make databases invaluable for most companies.

Google Fire Store is one of the standout cloud database services preferred by a large number of businesses today. It facilitates advanced data management and real-time functionality for comprehensive application development. Read on to gain a detailed insight into Google Fire store features and advantages.

Google Fire store, also known as Cloud Fire store is a part of the Google Firebase application development platform. It is fundamentally a cloud-hosted NoSQL database for storing and syncing data. Fire store can be directly accessed by mobile and web applications through native SDKs.

It enables users to avail options of Unity, Java, C++, Go, and Node.js SDKs and offers support for REST and RPC APIs. The Fire store database enables automatic scaling, enhanced performance, ease-of-use, and also provides a high level of reliability.

Firestore helps to sync data across multiple client applications with the use of real-time listeners. It uses the Cloud Identity, and Access Management features from Google for the process of authentication. Fire store performs data storage in the form of documents, with the documents being stored in collections.

Documents support a wide variety of data types, such as nested objects, numbers, and strings. Fire store enjoys integration with Google Cloud Platform and Google Firebase. Businesses prefer Firestore for the level of security and reliability it offers.

Fire store Key Features:

Automatic scaling

The fire store has been designed to scale automatically depending on user demand. It retains the same level of performance irrespective of database size. Fire store database size does not impact query time.

Serverless development

Client-side SDKs of Firestone handle networking and authentication while reducing the requirement for coding. Its backend security rules enable swift access to data and help you apply sophisticated validation logic on data.

Offline usage

Cloud Fire store from Google facilitates convenient offline usage through a robust database on users' devices. Offline data access ensures that applications run seamlessly even if the user gets disconnected from the Internet. Offline functionality can be availed by users across web, iOS, and Android platforms.

Datastore mode

Cloud Fire store offers Datastore API support. Developers do not need to make any changes to current Datastore applications. This feature helps users get uniform performance and benchmark stability while maintaining cost efficiency.

Robust query engine

Google Fire store has a sturdy and high-performance query engine. This critical feature helps developers execute complex queries against NoSQL data. It helps to do so without causing any compromises in performance. Users get greater flexibility when it comes to data structuring.

Robust security

Firestore from Google offers powerful security features for its users. It provides a convenient disaster recovery for protecting data at all times. Firestore also has features for both web and mobile platforms, including non-cascading rules and automatic data validation.

Multi-region replication

Firestore offers multi-region replication for data security. This feature breeds stability and helps to ensure availability in case of disasters. Automatic multi-regional replication of the Fire store helps to cut down any latency.

ACID transaction

ACID (atomicity, consistency, isolation, and durability) transaction is an essential feature of the Cloud Fire store from Google. This feature helps to support transactions. In the case of operational failures in a transaction, the transaction will fail.

Created for cloud-native applications

Fire store is developed for cloud-native applications. Web and mobile applications are among the workloads that cooperate with communications, retail catalogs, social user profiles, leaderboards, etc.

Fire store Advantages-:

- **Optimal data handling** – Cloud Fire store offers support for indexed queries, enabling users to combine sort and filter functionalities with the help of single queries. The advantages of ACID transactions are available across documents and collections.

- **Offline support** – Fire store provides offline support for the Android, web, and iOS platforms. It translates to your applications being updated almost in real time when any changes are made to the backend data.
- Integrates with Firebase / Google Cloud – Fire store is a combined project of Firebase and the Google Cloud Platform. It offers users the benefits of both services. Developers can enjoy the real-time database features of Firebase with Google Cloud's scalability.

- **Both serverless and cloud** – Cloud Fire store is serverless for facilitating a positive experience for developers, along with removing the need for server setup and data access management. Integration with Firebase and Google Cloud Platform facilitates better iterations, prototyping, offline support, and advanced synchronization.

- **Optimal scalability** – Cloud Fire store provides a great deal of scalability as it harnesses the robust Google Cloud Platform. Users can opt for horizontal scaling based on the load of an application. The service makes it easier for users to handle high volumes of complex data, which is stored across documents and sub-collections.

- **Pay-as-you-go pricing** – Google Cloud Fire store is considered to be a cost-effective option for developers. The more you scale, the more you pay. The lower your usage, the lower your costs.

5.2.5 Model progress Hud nsn:-

what is the Model progress Hud nsn:

The MB Progress HUD library is a popular open-source library for iOS applications that provides a versatile and customizable progress HUD (Heads-Up Display) for displaying progress indicators and other status information to users. It is written in Objective-C and can be integrated into Swift projects as well.

A simple widget wrapper to enable modal progress HUD (a modal progress indicator, HUD = Heads Up Display)

A fork of the original modal progress Hud by mmcc007 with Null Safety and additional features.

Features:

1- Activity Indicator:

Displays a spinning activity indicator to inform users that a task is in progress.

2- Progress Indicators:

Supports both determinate and indeterminate progress indicators. The determinate indicator shows a progress bar or pie chart, while the indeterminate indicator shows an indefinite spinner.

3- Customizable Appearance:

Allows customization of colors, sizes, and text to match the app's design.

4- Text Messages:

Displays simple text messages, such as loading messages or success/error notifications.

5- Custom Views:

Supports displaying custom views within the HUD, enabling developers to create unique loading screens or notifications.

6- Animations:

Provides smooth animations for showing and hiding the HUD.

5.3 Tools:

1- Python

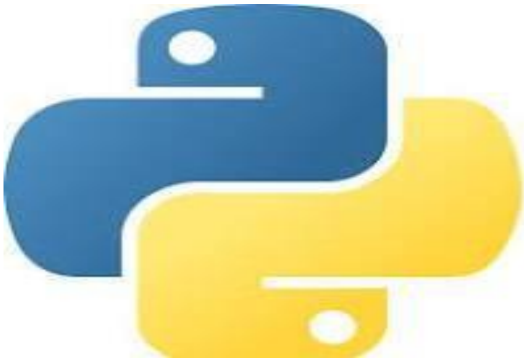


Figure 33 python

2- Flutter

Figure 34 flutter

3- Flask

Figure 35 flask

4- Dart

Figure 3c Dart

Chapter 6
References And Appendix

6.1 References:-

https://youtu.be/xQ2qiM0Zpr8?si=82ZXhGBy6aIHoWF

https://youtu.be/-a14O3Y2t0w?si=KhhUOsGvmK7z2Cwc

https://youtu.be/V9TYFe7hJy0?si=EqsgV2Li9wzCqCxd

https://youtu.be/QYKnX2VFkH8?si=qr5mg_2jusuWYZuC

https://scholar.google.com/

https://flutter.dev/

https://youtube.com/playlist?list=PLw6Y5u47CYq47oDw63bMqkq06fjuoK_GJ&si=HJ6h3dPWM1X7XuSt

https://youtu.be/99dnktVT-t4?si=tto7fbZmkXzlSu6C

https://youtu.be/tIlUSf0ojV4?si=fuJc9Lxkhor4Tz88

https://youtu.be/iZPICyh05wU?si=qK1_jJg-myGsyFlwhttps://youtu.be/iZPICyh05wU?si=qK1_jJg-myGsyFlwhttps://youtu.be/iZPICyh05wU?si=qK1_jJg-myGsyFlw

https://youtu.be/2qyH5LDxLdc?si=SgS0EH819Y6nftyU

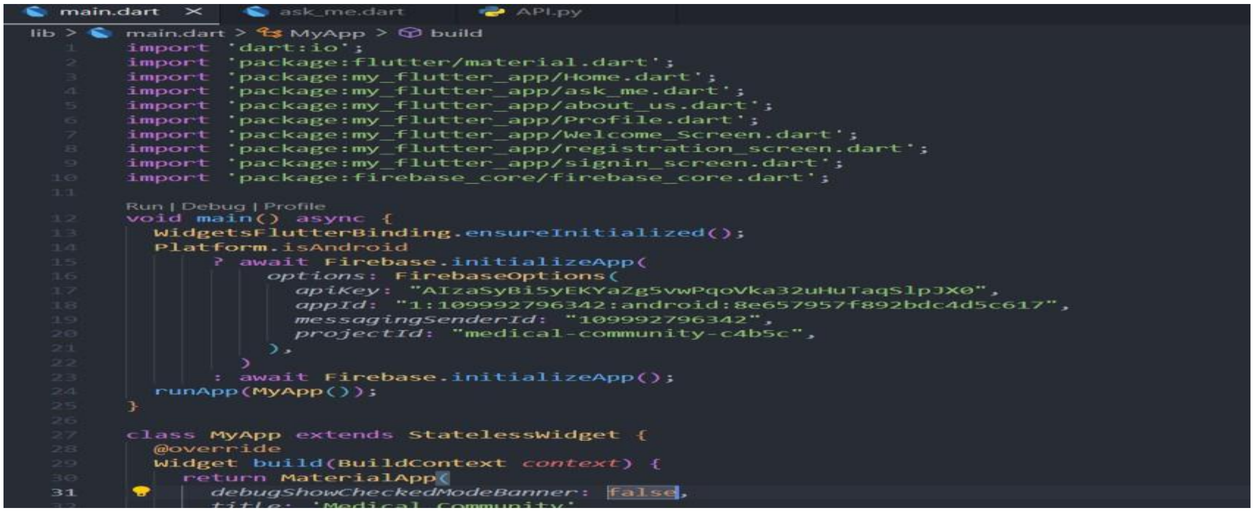
https://youtube.com/playlist?list=PL5zwAWCFLn1si9EScxEkfQkDhx1i4YW4s&si=3iWh5Ua1pduYmUer

https://www.kaggle.com/datasets/jackwin07/celiac-disease-coeliac-disease?resource=download

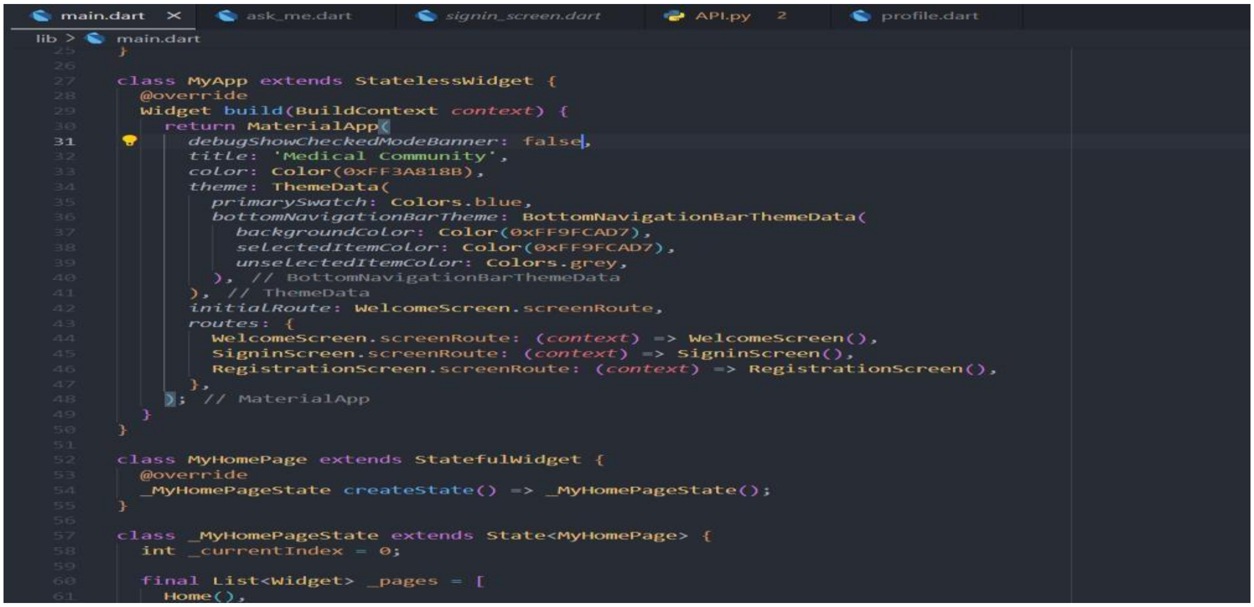
https://www.appicon.co/

6.2 Appendix:-

The code of main:

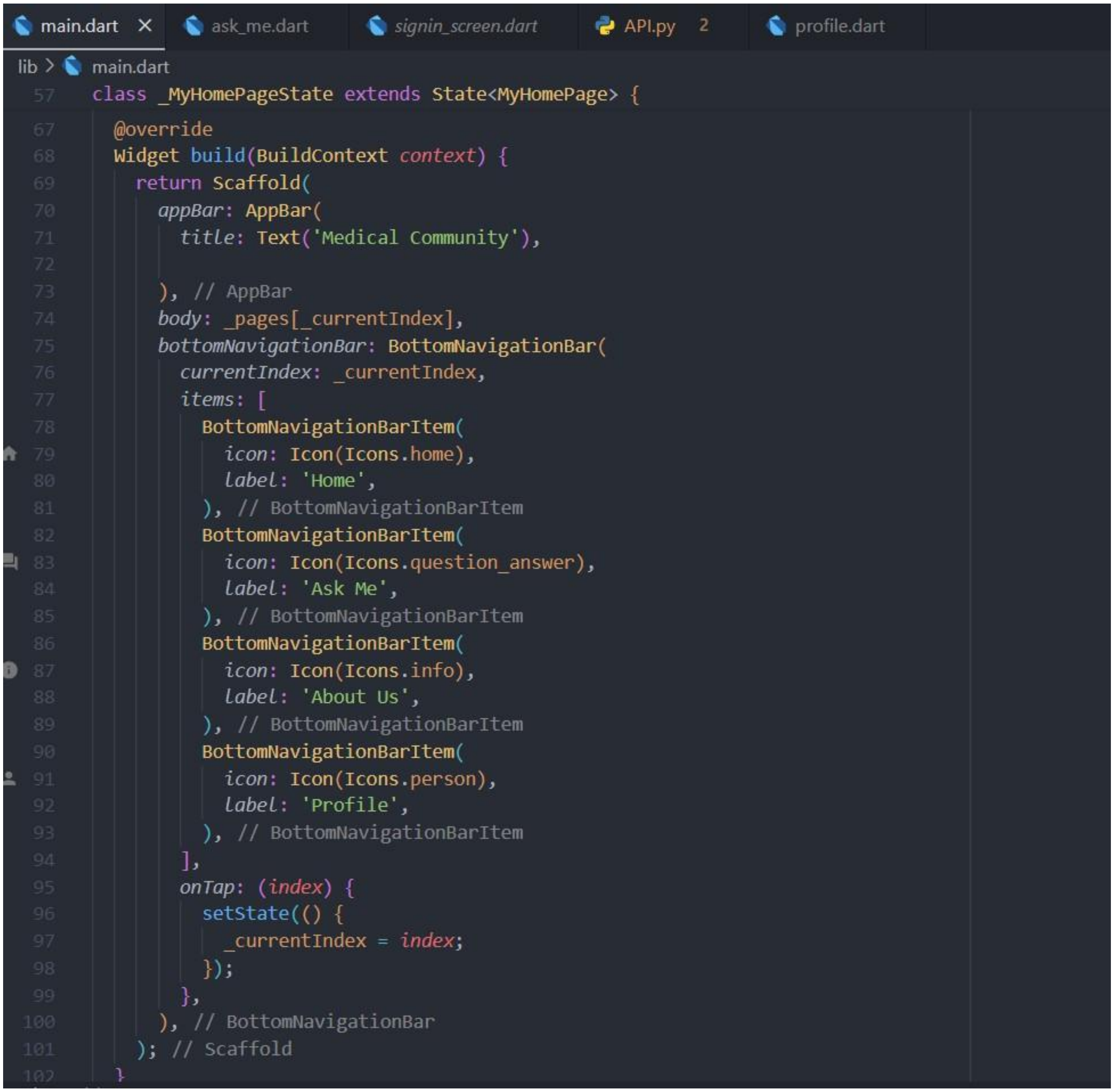


```dart
import 'dart:io';
import 'package:flutter/material.dart';
import 'package:my_flutter_app/Home.dart';
import 'package:my_flutter_app/ask_me.dart';
import 'package:my_flutter_app/about_us.dart';
import 'package:my_flutter_app/Profile.dart';
import 'package:my_flutter_app/Welcome_Screen.dart';
import 'package:my_flutter_app/registration_screen.dart';
import 'package:my_flutter_app/signin_screen.dart';
import 'package:firebase_core/firebase_core.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  Platform.isAndroid
        ? await Firebase.initializeApp(
            options: FirebaseOptions(
              apiKey: "AIzaSyBi5yEKYaZg5vwPqoVka32uHuTaqSlpJX0",
              appId: "1:109992796342:android:8e657957f892bdc4d5c617",
              messagingSenderId: "109992796342",
              projectId: "medical-community-c4b5c",
            ),
          )
        : await Firebase.initializeApp();
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Medical Community'
```



```dart
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Medical Community',
      color: Color(0xFF3A818B),
      theme: ThemeData(
        primarySwatch: Colors.blue,
        bottomNavigationBarTheme: BottomNavigationBarThemeData(
          backgroundColor: Color(0xFF9FCAD7),
          selectedItemColor: Color(0xFF9FCAD7),
          unselectedItemColor: Colors.grey,
        ), // BottomNavigationBarThemeData
      ), // ThemeData
      initialRoute: WelcomeScreen.screenRoute,
      routes: {
        WelcomeScreen.screenRoute: (context) => WelcomeScreen(),
        SigninScreen.screenRoute: (context) => SigninScreen(),
        RegistrationScreen.screenRoute: (context) => RegistrationScreen(),
      },
    ); // MaterialApp
  }
}

class MyHomePage extends StatefulWidget {
  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  int _currentIndex = 0;

  final List<Widget> _pages = [
    Home(),
```

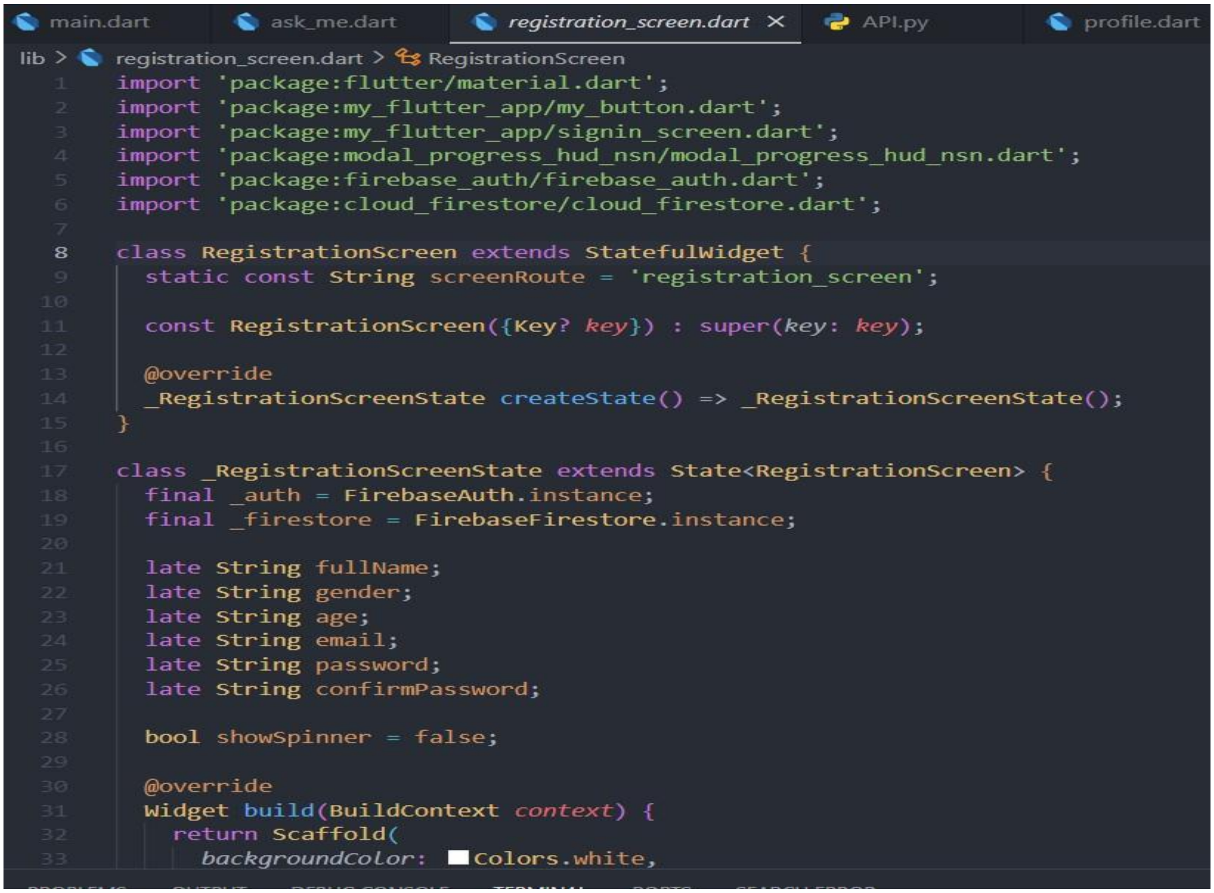
Figure 37 Code of main dart



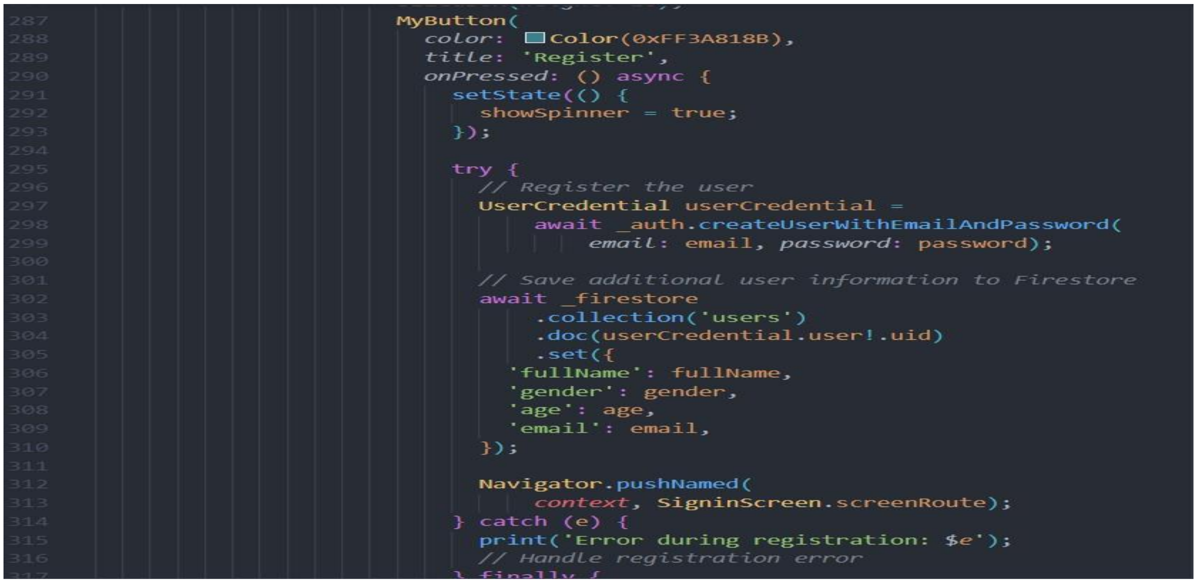
```
class _MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Medical Community'),

      ), // AppBar
      body: _pages[_currentIndex],
      bottomNavigationBar: BottomNavigationBar(
        currentIndex: _currentIndex,
        items: [
          BottomNavigationBarItem(
            icon: Icon(Icons.home),
            label: 'Home',
          ), // BottomNavigationBarItem
          BottomNavigationBarItem(
            icon: Icon(Icons.question_answer),
            label: 'Ask Me',
          ), // BottomNavigationBarItem
          BottomNavigationBarItem(
            icon: Icon(Icons.info),
            label: 'About Us',
          ), // BottomNavigationBarItem
          BottomNavigationBarItem(
            icon: Icon(Icons.person),
            label: 'Profile',
          ), // BottomNavigationBarItem
        ],
        onTap: (index) {
          setState(() {
            _currentIndex = index;
          });
        },
      ), // BottomNavigationBar
    ); // Scaffold
  }
```

Figure 38 Code of main dart

The code of register:



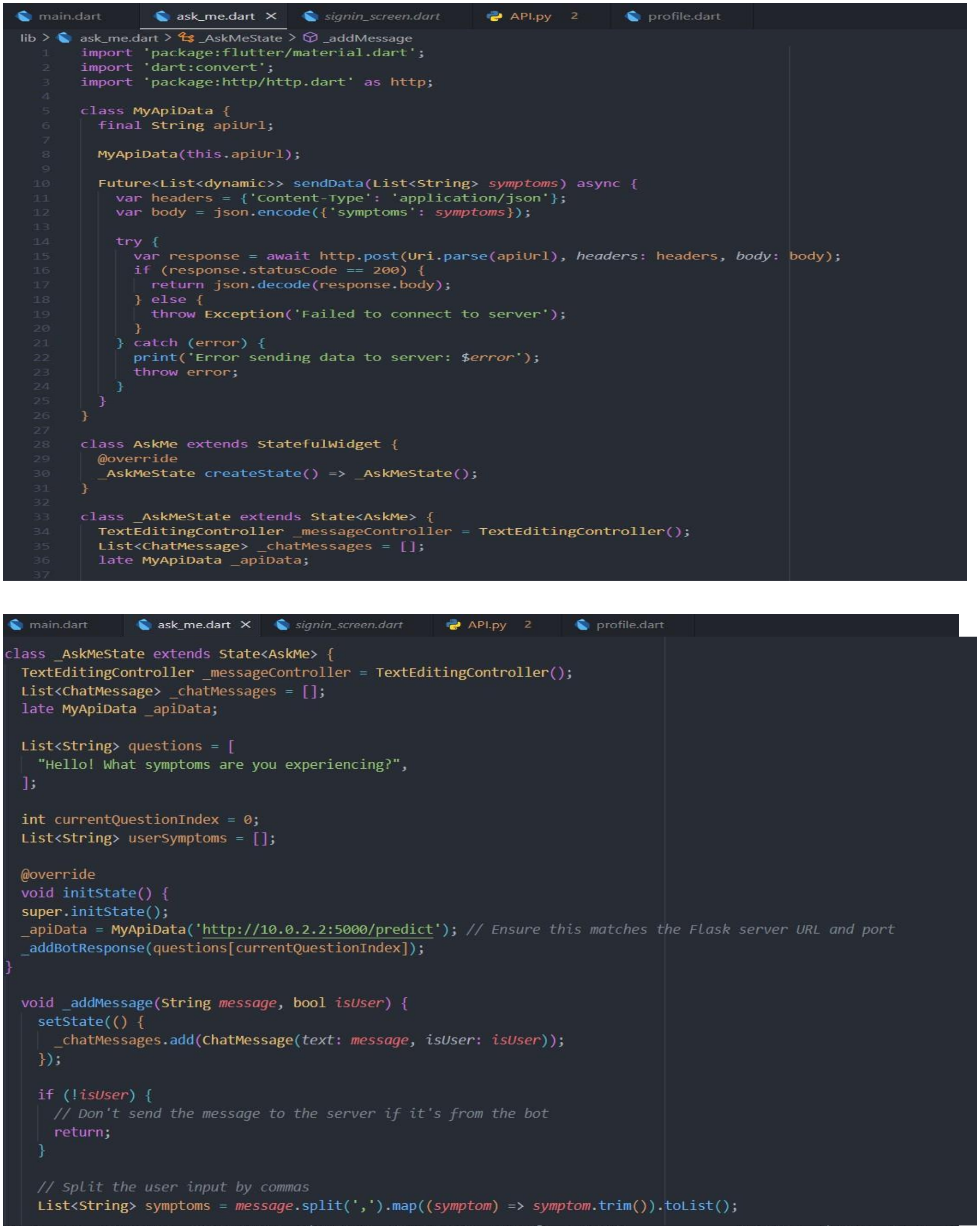
```dart
import 'package:flutter/material.dart';
import 'package:my_flutter_app/my_button.dart';
import 'package:my_flutter_app/signin_screen.dart';
import 'package:modal_progress_hud_nsn/modal_progress_hud_nsn.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:cloud_firestore/cloud_firestore.dart';

class RegistrationScreen extends StatefulWidget {
  static const String screenRoute = 'registration_screen';

  const RegistrationScreen({Key? key}) : super(key: key);

  @override
  _RegistrationScreenState createState() => _RegistrationScreenState();
}

class _RegistrationScreenState extends State<RegistrationScreen> {
  final _auth = FirebaseAuth.instance;
  final _firestore = FirebaseFirestore.instance;

  late String fullName;
  late String gender;
  late String age;
  late String email;
  late String password;
  late String confirmPassword;

  bool showSpinner = false;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: ⬜Colors.white,
```



```dart
MyButton(
  color: ■Color(0xFF3A818B),
  title: 'Register',
  onPressed: () async {
    setState(() {
      showSpinner = true;
    });

    try {
      // Register the user
      UserCredential userCredential =
          await _auth.createUserWithEmailAndPassword(
              email: email, password: password);

      // Save additional user information to Firestore
      await _firestore
          .collection('users')
          .doc(userCredential.user!.uid)
          .set({
        'fullName': fullName,
        'gender': gender,
        'age': age,
        'email': email,
      });

      Navigator.pushNamed(
          context, SigninScreen.screenRoute);
    } catch (e) {
      print('Error during registration: $e');
      // Handle registration error
    } finally {
```

Figure 3S Code of register

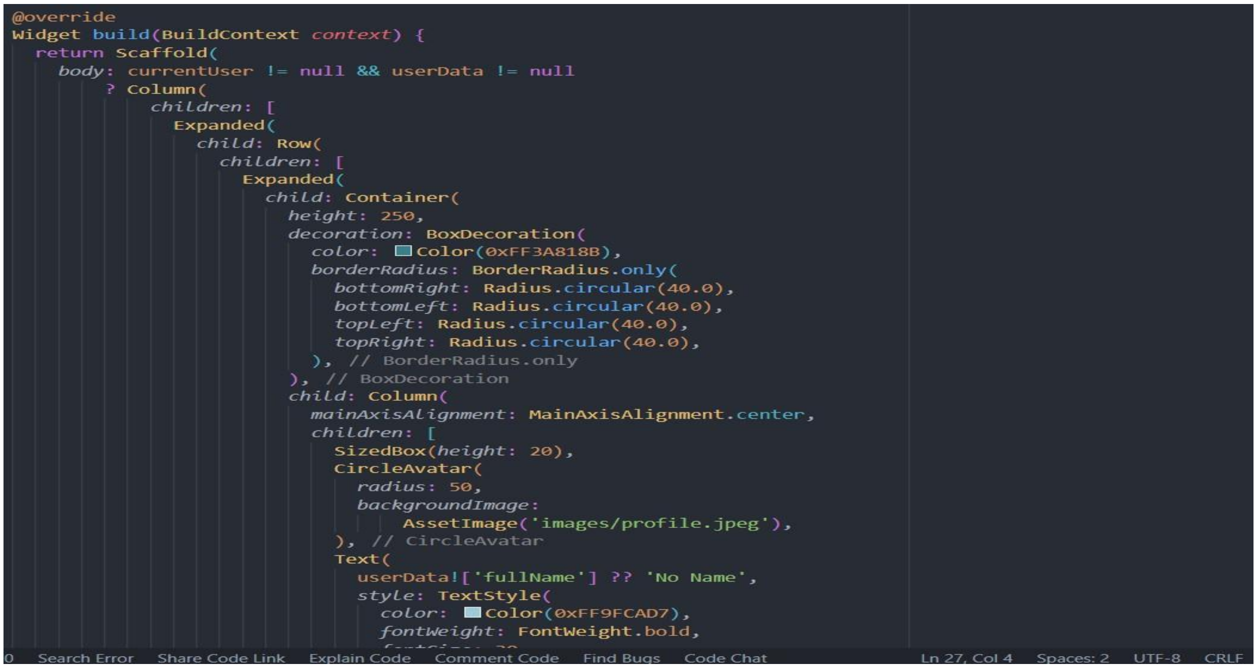
The code of Ask me:



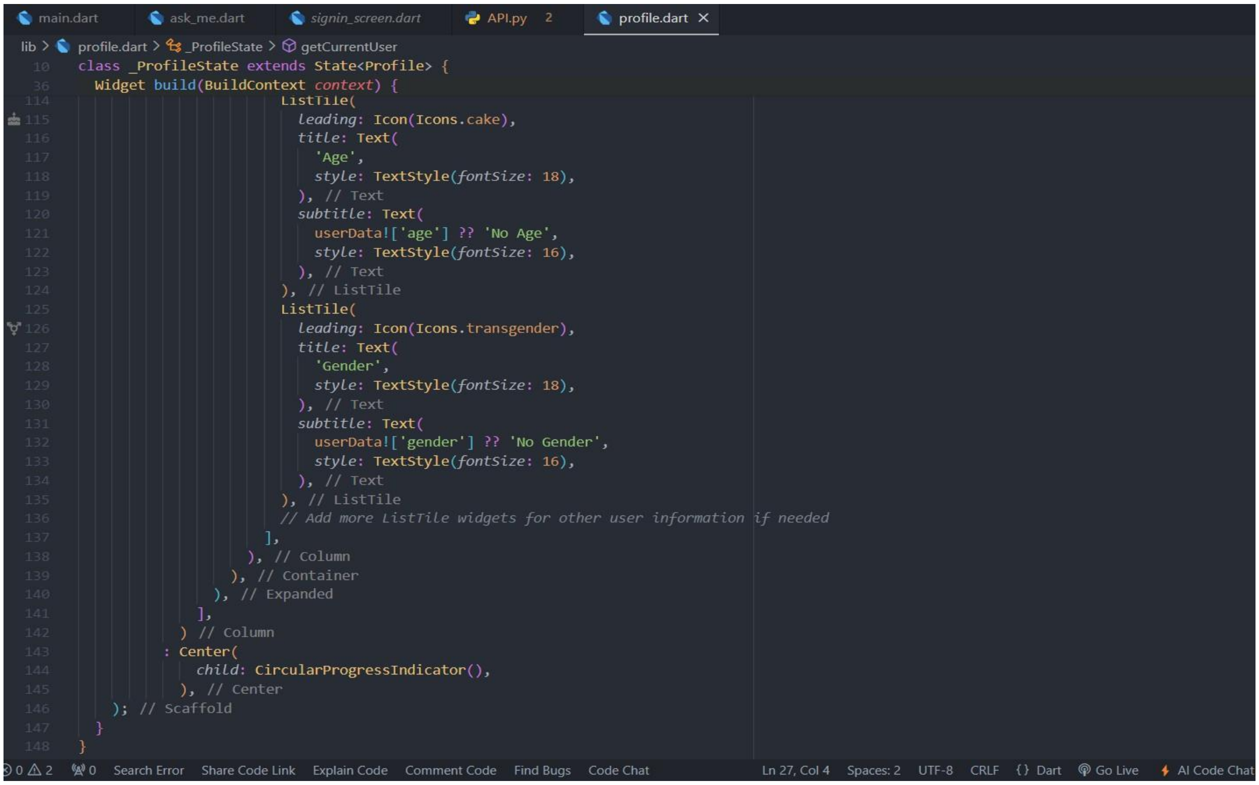
```
import 'package:flutter/material.dart';
import 'dart:convert';
import 'package:http/http.dart' as http;

class MyApiData {
  final String apiUrl;

  MyApiData(this.apiUrl);

  Future<List<dynamic>> sendData(List<String> symptoms) async {
    var headers = {'Content-Type': 'application/json'};
    var body = json.encode({'symptoms': symptoms});

    try {
      var response = await http.post(Uri.parse(apiUrl), headers: headers, body: body);
      if (response.statusCode == 200) {
        return json.decode(response.body);
      } else {
        throw Exception('Failed to connect to server');
      }
    } catch (error) {
      print('Error sending data to server: $error');
      throw error;
    }
  }
}

class AskMe extends StatefulWidget {
  @override
  _AskMeState createState() => _AskMeState();
}

class _AskMeState extends State<AskMe> {
  TextEditingController _messageController = TextEditingController();
  List<ChatMessage> _chatMessages = [];
  late MyApiData _apiData;

```

```
class _AskMeState extends State<AskMe> {
  TextEditingController _messageController = TextEditingController();
  List<ChatMessage> _chatMessages = [];
  late MyApiData _apiData;

  List<String> questions = [
    "Hello! What symptoms are you experiencing?",
  ];

  int currentQuestionIndex = 0;
  List<String> userSymptoms = [];

  @override
  void initState() {
  super.initState();
  _apiData = MyApiData('http://10.0.2.2:5000/predict'); // Ensure this matches the Flask server URL and port
  _addBotResponse(questions[currentQuestionIndex]);
}

  void _addMessage(String message, bool isUser) {
    setState(() {
      _chatMessages.add(ChatMessage(text: message, isUser: isUser));
    });

    if (!isUser) {
      // Don't send the message to the server if it's from the bot
      return;
    }

    // Split the user input by commas
    List<String> symptoms = message.split(',').map((symptom) => symptom.trim()).toList();
```

Figure 40 Code of ask me

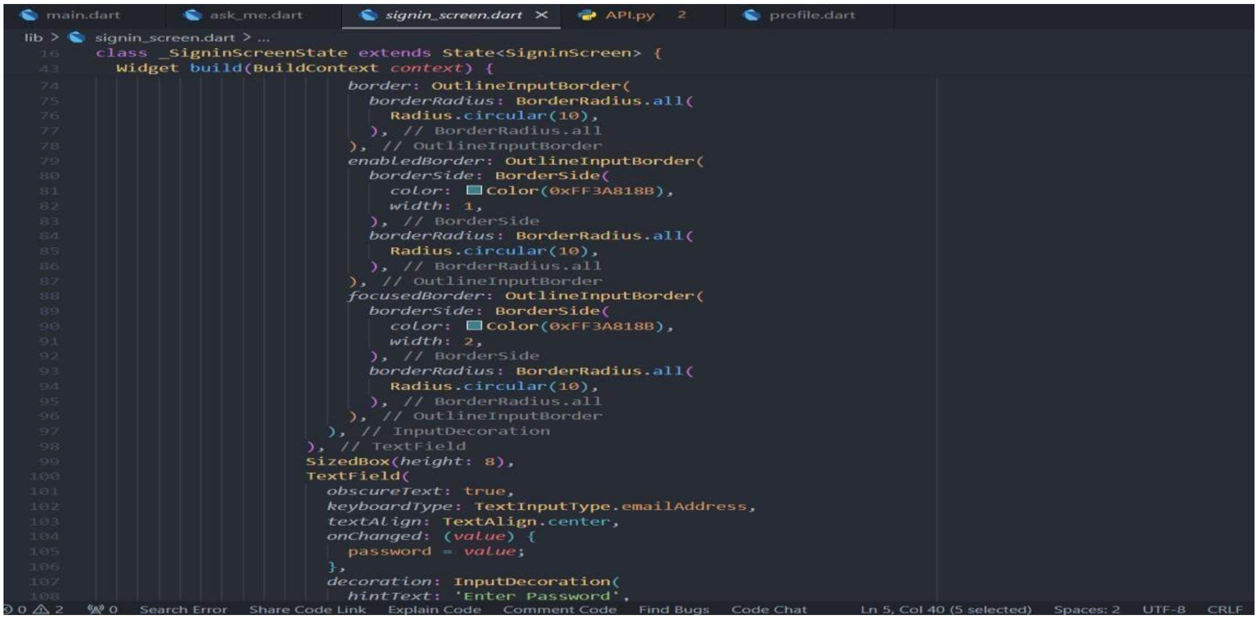


```
@override
Widget build(BuildContext context) {
  return Scaffold(
    body: currentUser != null && userData != null
        ? Column(
            children: [
              Expanded(
                child: Row(
                  children: [
                    Expanded(
                      child: Container(
                        height: 250,
                        decoration: BoxDecoration(
                          color: ▇Color(0xFF3A818B),
                          borderRadius: BorderRadius.only(
                            bottomRight: Radius.circular(40.0),
                            bottomLeft: Radius.circular(40.0),
                            topLeft: Radius.circular(40.0),
                            topRight: Radius.circular(40.0),
                          ), // BorderRadius.only
                        ), // BoxDecoration
                        child: Column(
                          mainAxisAlignment: MainAxisAlignment.center,
                          children: [
                            SizedBox(height: 20),
                            CircleAvatar(
                              radius: 50,
                              backgroundImage:
                                  AssetImage('images/profile.jpeg'),
                            ), // CircleAvatar
                            Text(
                              userData!['fullName'] ?? 'No Name',
                              style: TextStyle(
                                color: ▇Color(0xFF9FCAD7),
                                fontWeight: FontWeight.bold,
```

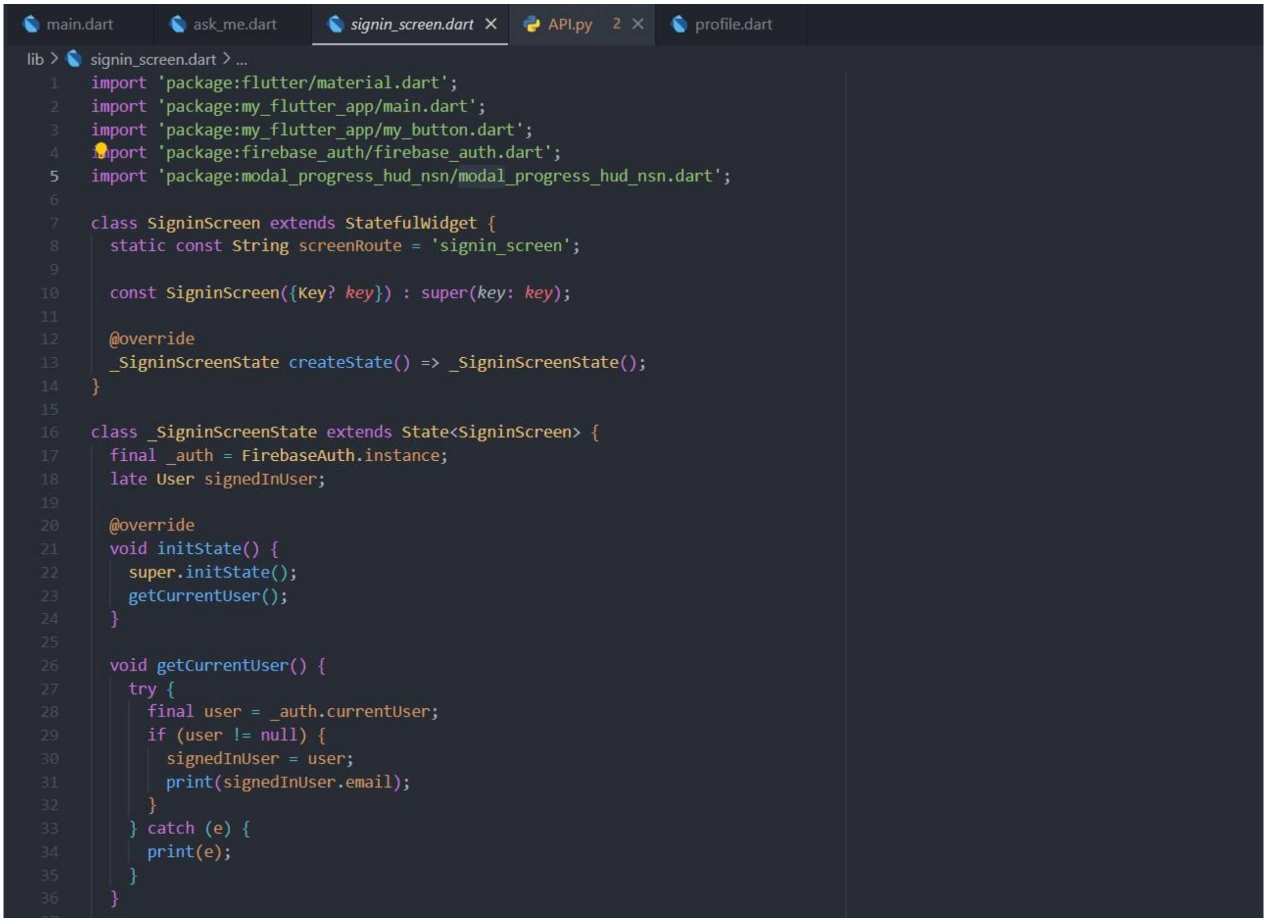


```
class _ProfileState extends State<Profile> {
  Widget build(BuildContext context) {
                        ListTile(
                          leading: Icon(Icons.cake),
                          title: Text(
                            'Age',
                            style: TextStyle(fontSize: 18),
                          ), // Text
                          subtitle: Text(
                            userData!['age'] ?? 'No Age',
                            style: TextStyle(fontSize: 16),
                          ), // Text
                        ), // ListTile
                        ListTile(
                          leading: Icon(Icons.transgender),
                          title: Text(
                            'Gender',
                            style: TextStyle(fontSize: 18),
                          ), // Text
                          subtitle: Text(
                            userData!['gender'] ?? 'No Gender',
                            style: TextStyle(fontSize: 16),
                          ), // Text
                        ), // ListTile
                        // Add more ListTile widgets for other user information if needed
                      ],
                    ), // Column
                  ), // Container
                ), // Expanded
              ],
            ) // Column
          : Center(
              child: CircularProgressIndicator(),
            ), // Center
    ); // Scaffold
  }
}
```

Figure 41 Code of profile



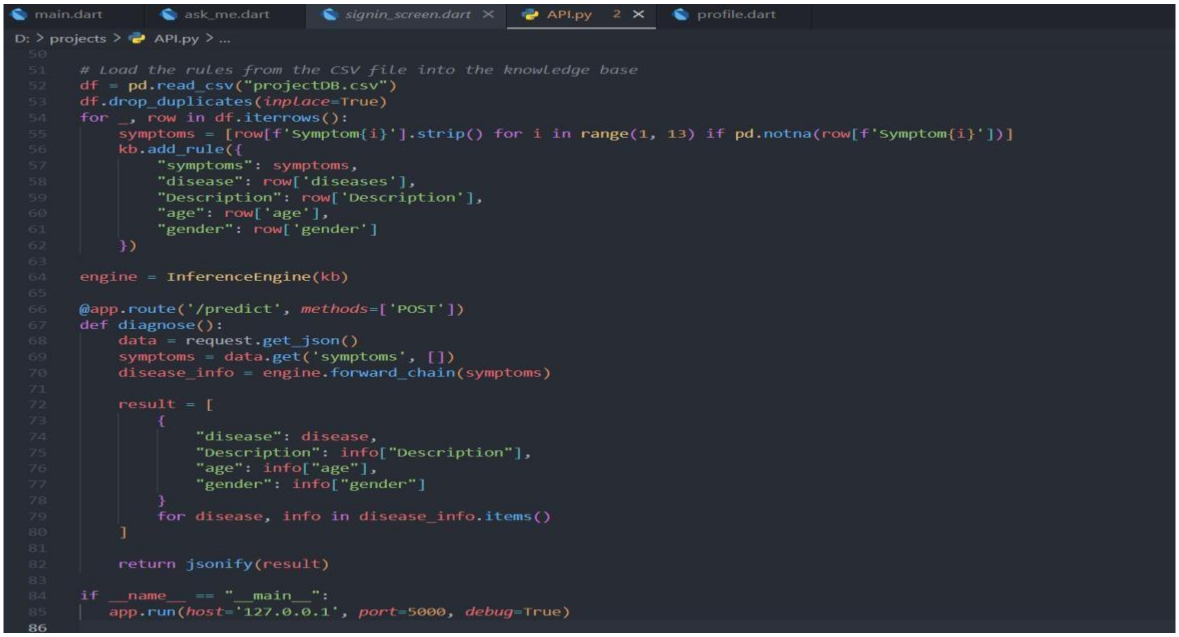
```
class _SigninScreenState extends State<SigninScreen> {
  Widget build(BuildContext context) {
                        border: OutlineInputBorder(
                          borderRadius: BorderRadius.all(
                            Radius.circular(10),
                          ), // BorderRadius.all
                        ), // OutlineInputBorder
                        enabledBorder: OutlineInputBorder(
                          borderSide: BorderSide(
                            color: Color(0xFF3A818B),
                            width: 1,
                          ), // BorderSide
                          borderRadius: BorderRadius.all(
                            Radius.circular(10),
                          ), // BorderRadius.all
                        ), // OutlineInputBorder
                        focusedBorder: OutlineInputBorder(
                          borderSide: BorderSide(
                            color: Color(0xFF3A818B),
                            width: 2,
                          ), // BorderSide
                          borderRadius: BorderRadius.all(
                            Radius.circular(10),
                          ), // BorderRadius.all
                        ), // OutlineInputBorder
                      ), // InputDecoration
                    ), // TextField
                    SizedBox(height: 8),
                    TextField(
                      obscureText: true,
                      keyboardType: TextInputType.emailAddress,
                      textAlign: TextAlign.center,
                      onChanged: (value) {
                        password = value;
                      },
                      decoration: InputDecoration(
                        hintText: 'Enter Password',
```



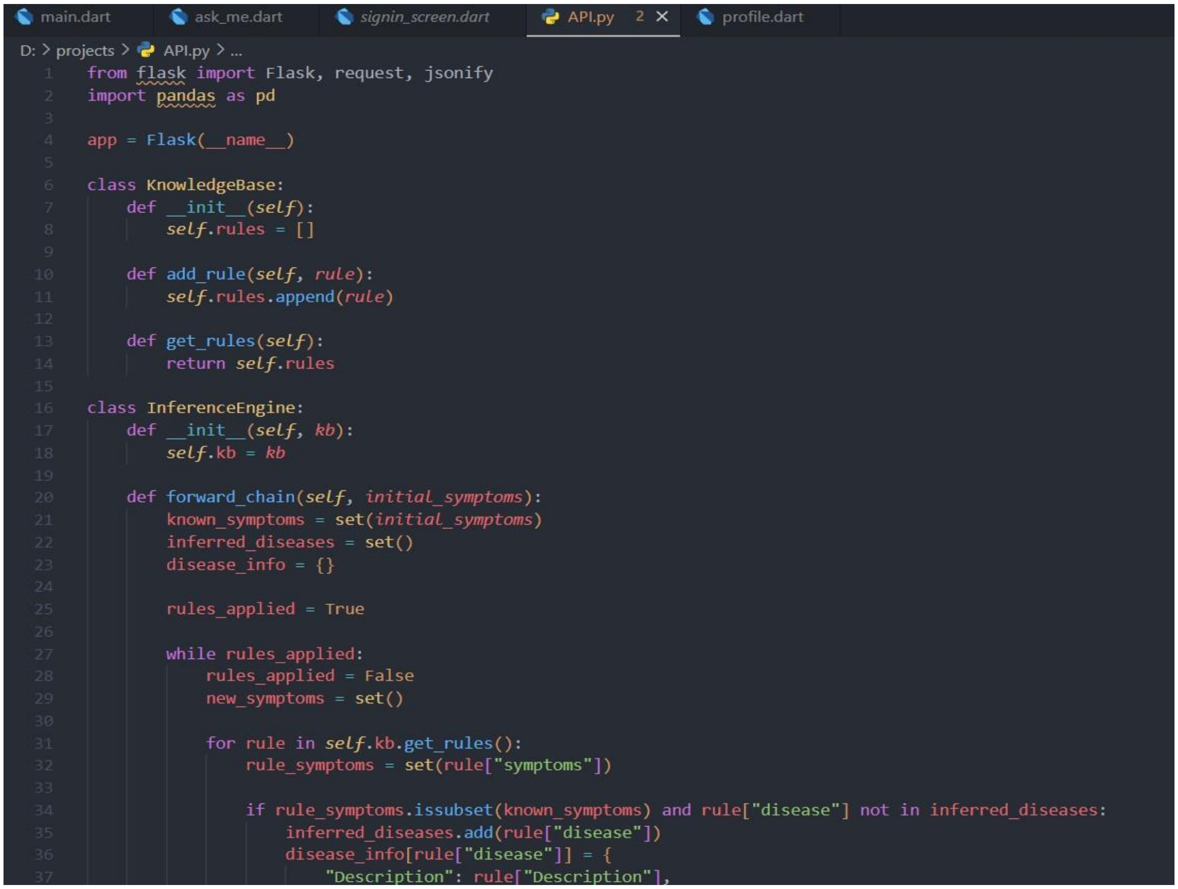
```
import 'package:flutter/material.dart';
import 'package:my_flutter_app/main.dart';
import 'package:my_flutter_app/my_button.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:modal_progress_hud_nsn/modal_progress_hud_nsn.dart';

class SigninScreen extends StatefulWidget {
  static const String screenRoute = 'signin_screen';

  const SigninScreen({Key? key}) : super(key: key);

  @override
  _SigninScreenState createState() => _SigninScreenState();
}

class _SigninScreenState extends State<SigninScreen> {
  final _auth = FirebaseAuth.instance;
  late User signedInUser;

  @override
  void initState() {
    super.initState();
    getCurrentUser();
  }

  void getCurrentUser() {
    try {
      final user = _auth.currentUser;
      if (user != null) {
        signedInUser = user;
        print(signedInUser.email);
      }
    } catch (e) {
      print(e);
    }
  }
```

Figure 38 Code of sign in

API Flask



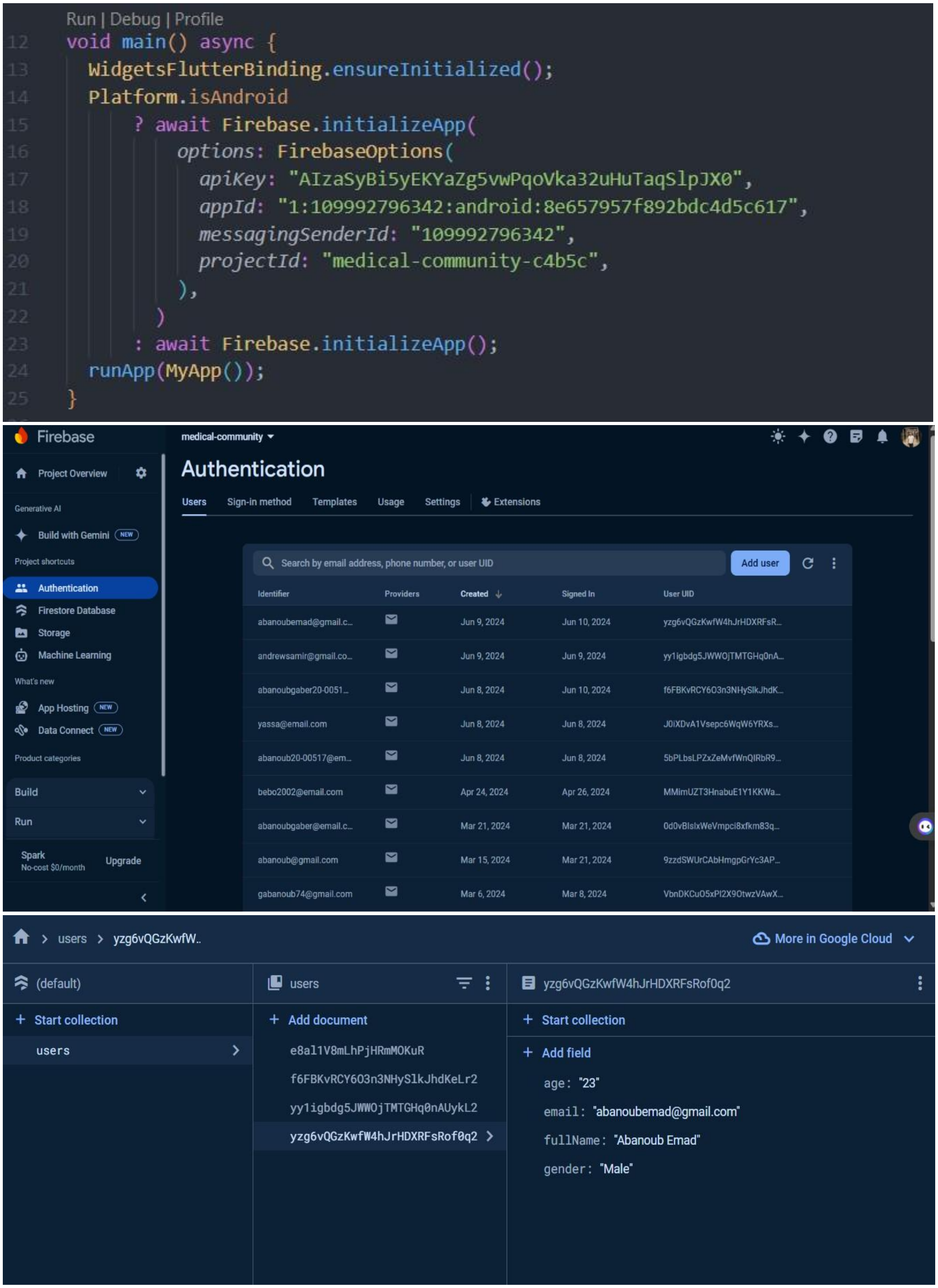
```python

# Load the rules from the CSV file into the knowledge base
df = pd.read_csv("projectDB.csv")
df.drop_duplicates(inplace=True)
for _, row in df.iterrows():
    symptoms = [row[f'Symptom{i}'].strip() for i in range(1, 13) if pd.notna(row[f'Symptom{i}'])]
    kb.add_rule({
        "symptoms": symptoms,
        "disease": row['diseases'],
        "Description": row['Description'],
        "age": row['age'],
        "gender": row['gender']
    })

engine = InferenceEngine(kb)

@app.route('/predict', methods=['POST'])
def diagnose():
    data = request.get_json()
    symptoms = data.get('symptoms', [])
    disease_info = engine.forward_chain(symptoms)

    result = [
        {
            "disease": disease,
            "Description": info["Description"],
            "age": info["age"],
            "gender": info["gender"]
        }
        for disease, info in disease_info.items()
    ]

    return jsonify(result)

if __name__ == "__main__":
   app.run(host='127.0.0.1', port=5000, debug=True)

```



```python
from flask import Flask, request, jsonify
import pandas as pd

app = Flask(__name__)

class KnowledgeBase:
    def __init__(self):
        self.rules = []

    def add_rule(self, rule):
        self.rules.append(rule)

    def get_rules(self):
        return self.rules

class InferenceEngine:
    def __init__(self, kb):
        self.kb = kb

    def forward_chain(self, initial_symptoms):
        known_symptoms = set(initial_symptoms)
        inferred_diseases = set()
        disease_info = {}

        rules_applied = True

        while rules_applied:
            rules_applied = False
            new_symptoms = set()

            for rule in self.kb.get_rules():
                rule_symptoms = set(rule["symptoms"])

                if rule_symptoms.issubset(known_symptoms) and rule["disease"] not in inferred_diseases:
                    inferred_diseases.add(rule["disease"])
                    disease_info[rule["disease"]] = {
                        "Description": rule["Description"],
```

Figure 35 Code of Api flak & expert system

Firebase and firestore



```
Run | Debug | Profile
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  Platform.isAndroid
      ? await Firebase.initializeApp(
          options: FirebaseOptions(
            apiKey: "AIzaSyBi5yEKYaZg5vwPqoVka32uHuTaqSlpJX0",
            appId: "1:109992796342:android:8e657957f892bdc4d5c617",
            messagingSenderId: "109992796342",
            projectId: "medical-community-c4b5c",
          ),
        )
      : await Firebase.initializeApp();
  runApp(MyApp());
}
```

Authentication

Identifier	Providers	Created	Signed In	User UID
abanoubemad@gmail.c...	✉	Jun 9, 2024	Jun 10, 2024	yzg6vQGzKwfW4hJrHDXRFsR...
andrewsamir@gmail.co...	✉	Jun 9, 2024	Jun 9, 2024	yy1igbdg5JWWOjTMTGHq0nA...
abanoubgaber20-0051...	✉	Jun 8, 2024	Jun 10, 2024	f6FBKvRCY6O3n3NHySlkJhdK...
yassa@email.com	✉	Jun 8, 2024	Jun 8, 2024	J0iXDvA1Vsepc6WqW6YRXs...
abanoub20-00517@em...	✉	Jun 8, 2024	Jun 8, 2024	5bPLbsLPZxZeMvfWnQIRbR9...
bebo2002@email.com	✉	Apr 24, 2024	Apr 26, 2024	MMimUZT3HnabuE1Y1KKWa...
abanoubgaber@email.c...	✉	Mar 21, 2024	Mar 21, 2024	0d0vBIslxWeVmpci8xfkm83q...
abanoub@gmail.com	✉	Mar 15, 2024	Mar 21, 2024	9zzdSWUrCAbHmgpGrYc3AP...
gabanoub74@gmail.com	✉	Mar 6, 2024	Mar 8, 2024	VbnDKCuO5xPI2X9OtwzVAwX...

users:
- e8al1V8mLhPjHRmMOKuR
- f6FBKvRCY6O3n3NHySlkJhdKeLr2
- yy1igbdg5JWWOjTMTGHq0nAUykL2
- yzg6vQGzKwfW4hJrHDXRFsRof0q2

yzg6vQGzKwfW4hJrHDXRFsRof0q2

age: "23"
email: "abanoubemad@gmail.com"
fullName: "Abanoub Emad"
gender: "Male"

Figure 40 Code of Firebase (Authentication)