**Chapter 1**

**Coding and Tools**

**1.1 HTML**

**1.1.1 WHAT is HTML?**

HTML5 is the latest version of Hypertext Markup Language, the code that describes web pages. It's actually three kinds of code: HTML, which provides the structure; Cascading Style Sheets (CSS), which take care of presentation; and JavaScript, which makes things happen.

**1.1.2 What's great about HTML5?**

HTML5 has been designed to deliver almost everything you'd want to do online without requiring additional software such as browser plugins. It does everything from animation to apps, music to movies, and can also be used to build incredibly complicated applications that run in your browser.

There's more. HTML5 isn't proprietary, so you don't need to pay royalties to use it. It's also cross-platform, which means it doesn't care whether you're using a tablet or a smartphone, a netbook, notebook or ultrabook or a Smart TV: if your browser supports HTML5, it should work flawlessly. Inevitably, it's a bit more complicated than that. More about that in a moment.

**1.1.3 What does HTML5 do?**

We've come a long way since HTML could barely handle a simple page layout. HTML5 can be used to write web applications that still work when you're not connected to the net; to tell websites where you are physically located; to handle high definition video; and to deliver extraordinary graphics.

**1.1.4 What tools do I need to create HTML5 websites?**

Purists would say that you don't need any tools - you can write HTML5 code with a biro and a bit of paper if you wish - but others prefer tools such as Adobe's Dreamweaver, which gained an HTML5 pack in 2010.

Adobe is also readying Edge, a dedicated HTML5 animation tool that promises to make it easy to create Flash-style animation.

**1.1.5 HTML5 Features**

Web languages need regular upgrades in order to stay current and solve new problem faced by web developers

HTML5 is the latest version of HTML

Below are some HTML5 features you will encounter as you learn with Codecademy.

Video:

The video element allows you to easily stream video from a website.

<video width="450px" height="350px" controls>

<source src="video-url.mp4"

<"type="video/mp4>

<video/>

**Figure**

Figure element can be used to display visual content, such as photos, illustration, diagrams or code snippets.

<figure class="gallery-item">

<img src="image-1.png">

</figure>

<figure class="gallery-item">

<img src="image-2.png">

</figure>

**Section**

Section element, like dives, can be used to organize webpage content into thematic groups.

<section class="contact-form >

<h2> Contact Us </h2>

<form>

</form>

</section>

**1.2 CSS**

**1.2.1 What is CSS?**

CSS stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once. External style sheets are stored in CSS files.

**1.2.2 CSS language Features?**

* CSS Plays an important role with CSS, you simply have to define a recurring style of the element once and use it several times
* Web designers need to use a few line of code per page to improve site speed.
* it is less complex and therefore the effort is greatly reduced.
* Helps create spontaneous and consistent changes.
* Device-friendly CSS changes, with people employing a variety of smart devices to access the web, there are requirements for responsive web design.

**1.2.3 Why Use CSS?**

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

**CSS Example**

**Body{background-color: lightblue ;}**

**h1 {**

**color: white;**

**text-align: center;**

**}**

**1.2.4 CSS solved a Big Problem!**

HTML was NEVER intended to contain tags for formatting a web page!

HTML was created to describe the content of a web page, like:

<h1> this is a heading </h1>

<p> this is a paragraph. </p>

When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS.

CSS removed the style formatting from the HTML page!

**1.2.5 CSS saves a lot of Work!**

The style definitions are normally saved in external .css files.

With an external style sheet file, you can change the look of an entire website by changing just one file!

**1.2.6 CSS syntax**

OBJ

The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

**1.2.7 CSS Selectors:**

CSS selectors are used to "find" (or select) the HTML elements you want to style.

We can divide CSS selectors into five categories:

Simple selectors (select elements based on name, id, class).

Combinator selectors (select elements based on a specific relationship between them).

Pseudo-class selectors (select elements based on a certain state).

Pseudo-elements selectors (select and style a part of an element).

Attribute selectors (select elements based on an attribute or attribute value).

**1.2.8 The CSS element Selector**

The element selector selects HTML elements based on the element name.

**Example**

Here, all <p> elements on the page will be center-aligned, with a red text color:

P

{

text-align: center;

 color: red;

}

**1.2.8.1 The CSS id Selector**

The id selector uses the id attribute of an HTML element to select a specific element.The id of an element is unique within a page, so the id selector is used to select one unique element! To select an element with a specific id, write a hash (#) character, followed by the id of the element.

**1.2.8.2 The CSS CLASS Selector**

The class selector selects HTML elements with a specific class attribute. To select elements with a specific class, write a period (.) character, followed by the class name.

**1.2.9 There Ways to Insert CSS**

There are three ways of inserting a style sheet:

* External CSS
* Internal CSS
* Inline CSS

**1.2.9.1 External CSS**

With an external style sheet, you can change the look of an entire website by changing just one file!

Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.

**1.2.9.2 Internal CSS:**

An internal style sheet may be used if one single HTML page has a unique style.

The internal style is defined inside the <style> element, inside the head section.

**1.2.9.3 Inline CSS**

An inline style may be used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

**1.3 PHP**

**1.3.1 What is PHP?**

PHP is an open-source, server-side, HTML-embedded **scripting language** that is used to develop **Static or Dynamic websites** or Web applications.

PHP stands for Hypertext Pre-processor, that earlier stood for Personal Home Pages.

PHP scripts can only be interpreted on a server that has PHP installed.

The client computers accessing the PHP scripts require a web browser only.

A PHP file contains PHP tags and ends with the extension "php".

**1.3.2 How PHP came into being:**

PHP was conceived sometime in the fall of 1994 by [Rasmus Lerdorf](mailto:rasmus@php.net" \t "_top). Early non-released versions were used on his home page to keep track of who was looking at his online resume.

The first version used by others was available sometime in early 1995 and was known as the Personal Home Page Tools. It consisted of a very simplistic parser engine that only understood a few special macros and a number of utilities that were in common use on home pages back then.

A guestbook, a counter and some other stuff. The parser was rewritten in mid-1995 and named PHP/FI Version 2. The FI came from another package Rasmus had written which interpreted html form data. He combined the Personal Home Page tools scripts with the Form Interpreter and added mSQL support and PHP/FI was born. PHP/FI grew at an amazing pace and people started contributing code to it.

It is difficult to give any hard statistics, but it is estimated that by late 1996 PHP/FI was in use on at least 15,000 web sites around the world. By mid-1997 this number had grown to over 50,000. Mid-1997 also saw a change in the development of PHP. It changed from being Rasmus' own pet project that a handful of people had contributed to, to being a much more organized team effort.

The parser was rewritten from scratch by Zeev Suraski and Andi Gutmans and this new parser formed the basis for PHP Version 3. A lot of the utility code from PHP/FI was ported over to PHP3 and a lot of it was completely rewritten.

Today (end-1999) either PHP/FI or PHP3 ships with a number of commercial products such as C2's StrongHold web server and RedHat Linux. A conservative estimate based on an extrapolation from numbers provided by [NetCraft](http://www.netcraft.com/" \t "_top) (see also [Netcraft Web Server Survey](http://www.netcraft.com/survey/" \t "_top)) would be that PHP is in use on over 1,000,000 sites around the world. To put that in perspective, that is more sites than run Netscape's flagship Enterprise server on the Internet.

Also as of this writing, work is underway on the next generation of PHP, which will utilize the powerful [Zend](http://www.zend.com/" \t "_top) scripting engine to deliver higher performance, and will also support running under webservers other than Apache as a native server module.

**1.3.3 What does PHP offer, and why not use another language?**

* PHP is **open source and free.**
* Short learning curve compared to other languages such as JSP, ASP etc.
* Large community document
* Most web hosting servers support PHP by default unlike other languages such as ASP that need IIS. This makes PHP a cost effective choice.
* PHP is regular updated to keep abreast with the latest technology trends.
* Other benefit that you get with PHP is that it’s a **server side scripting language**; this means you only need to install it on the server and client computers requesting for resources from the server do not need to have PHP installed; only a web browser would be enough.
* PHP has **in built support for working hand in hand with MySQL**; this doesn’t mean you can’t use PHP with other database management systems. You can still use PHP with
  + Postgres
  + Oracle
  + MS[SQL](https://www.guru99.com/sql.html)Server
  + ODBC etc.
* PHP is **cross platform;** this means you can deploy your application on a number of different operating systems such as windows, Linux, Mac OS etc.

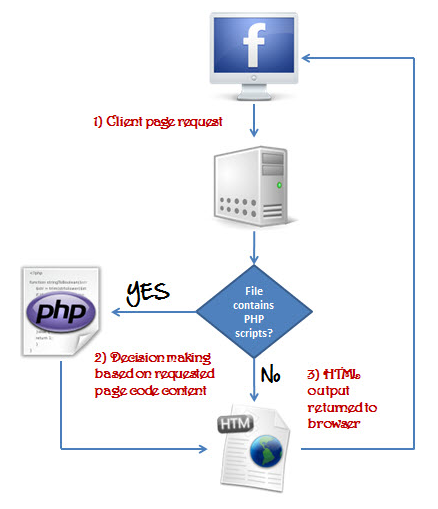
**1.3.4 Syntactical example:**

****

A PHP file can also contain tags such as HTML and client side scripts such as JavaScript.

* **HTML** **is an added advantage** when learning PHP Language. You can even learn PHP without knowing HTML but it’s recommended you at least know the basics of HTML.
* **Database management systems** DBMS for database powered applications.
* For more advanced topics such as interactive applications and web services, you will need **JavaScript and XML**.

The flowchart diagram shown below illustrates the basic architecture of a PHP web application and how the server handles the requests.



Also notice how this is different from a CGI script written in other languages like Perl or C -- instead of writing a program with lots of commands to output HTML, you write an HTML script with a some embedded code to do something (in this case, output some text). The PHP code is enclosed in special [start and end tags](https://ifj.edu.pl/private/krawczyk/php/language.basic-syntax.html#language.basic-syntax.phpmode) that allow you to jump into and out of PHP mode.

What distinguishes PHP from something like client-side Javascript is that the code is executed on the server. If you were to have a script similar to the above on your server, the client would receive the results of running that script, with no way of determining what the underlying code may be. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve.

## So what is a Scripting Language?

A script is a set of programming instructions that is interpreted at runtime.

A scripting language is a language that interprets scripts at runtime. Scripts are usually embedded into other software environments.

The purpose of the scripts is usually to enhance the performance or perform routine tasks for an application.

Server side scripts are interpreted on the server while client side scripts are interpreted by the client application.

PHP is a server side script that is interpreted on the server while[JavaScript](https://www.guru99.com/interactive-javascript-tutorials.html)is an example of a client side script that is interpreted by the client browser. Both PHP and JavaScript can be embedded into HTML pages.

**1.3.5 PHP File Extensions**

*File extension and Tags* In order for the **server** to **identify** our **PHP** **files** and **scripts**, we must **save** the **file** with the **“.php” extension**. Older PHP file extensions include

* .phtml
* .php3
* .php4
* .php5
* .phps

PHP was designed to work with HTML, and as such, it can be embedded into the HTML code.



You can create PHP files without any html tags and that is called a **Pure PHP** file.

The server interprets the PHP code and outputs the results as HTML code to the web browsers.

In order for the server to identify the PHP code from the HTML code, we must always enclose the PHP code in PHP tags.

A PHP tag starts with the less than symbol followed by the question mark and then the words “php”.

PHP is a case sensitive language, “VAR” is not the same as “var”.

The PHP tags themselves are not case-sensitive, but it is strongly recommended that we use lower case letter. The code below illustrates the above point.



We refer to the PHP lines of code as **statements**. PHP statements end with a semi colon (;). If you only have one statement, you can omit the semi colon. If you have more than one statement, then you must end each line with a semi colon. For the sake of consistency, it is recommended that you always end your statement(s) with a semi colon.  PHP scripts are executed on the server. The output is returned in form of HTML.

## 1.3.6 Market share:

In terms of market share, there are over 20 million websites and application on the internet developed using PHP scripting language, this may be attributed to the points raised above.

## 1.3.7 PHP VS Asp.Net VS JSP VS CFML

## [ASP](https://www.guru99.com/asp-net-tutorial.html) – Active Server Pages, [JSP](https://www.guru99.com/jsp-tutorial.html)– Java Server Pages, CFML – Cold Fusion Markup language The table below compares the various server side scripting languages with PHP

## 

**1.3.8 Dynamic** VS **Static websites:**

Websites are separated into two different types: static and dynamic. Static websites are ones that are fixed and display the same content for every user, usually written exclusively in HTML. A dynamic website, on the other hand, is one that can display different content and provide user interaction, by making use of advanced programming and databases in addition to HTML. Static websites are easier to create, while dynamic websites require more work.

Before getting into details about each website type, understand how the Internet goes about serving up websites in the first place. Internet communication involves a server and a web browser.

To establish a connection between the two, a set of rules called Hypertext Transfer Protocol (HTTP) is used. Simply put, the web browser transmits an HTTP request to the server, and the server then replies with an HTTP response along with the requested webpage in HTML.

### 1.3.8.1 What is a static website?

Static [websites usually come with a fixed number of pages that have a specific layout.](https://wpamelia.com/website-layouts/) When the page runs on a browser, the content is literally static and doesn’t change in response to user actions. A static website is usually created with HTML and CSS in simple text editors like Notepad.

If you need a [website smaller than three pages,](https://wpamelia.com/single-page-website/) opting for a static website is the proper choice. Building it doesn’t take as much time or effort as in the case of dynamic websites. If the pages of your website must look different, the HTML code can easily be duplicated on each of these pages, containing the necessary changes.

Even though the website will display the same thing with no intricate navigation details, static websites don’t need to feature just plain text. In fact, you can use various multimedia elements and videos. An HTML website can look beautiful, but the page’s [source code](https://wpamelia.com/javascript-ide/) won’t change, no matter what actions a user takes on it.

### 1.3.8.2 What is a Dynamic website?

Compared to static websites, which are purely informational, a dynamic website is more functional. It allows users to interact with the information that is listed on the page. Of course, that requires utilizing more than just HTML code.

Static websites use only client-side HTML and CSS code while dynamic websites rely on both client-side and server-side scripting languages such as JavaScript, PHP, or ASP. When a user accesses a dynamic website, the site can be changed through code that is run in the browser and/or on the server. The end result is the same as that on a static website: an HTML page displayed on the web browser.

To generate dynamic content, such websites use a combination of server-side and client-side scripting. Client-side scripting refers to code that is executed by the browser, usually with JavaScript. Meanwhile, server-side scripting refers to code that is executed by the server (before the content is sent to the user’s browser).

### 1.3.8.3 Static VS Dynamic:

* **Meaning**

The word static refers to something that is fixed, that doesn’t move or change in any way. This is enough to understand what a static page is all about. No elements on this page are changed when accessing it.

Static websites are basic pages that require simple code and design elements to create. “Static” also refers to the website being fixed in terms of page numbers. A fixed number of pages are delivered just the way it is designed and stored.

Conversely, the word dynamic refers to elements that are continuously changing, interactive, and functional. Instead of being simply informational, dynamic websites include aspects that are characterized by interactivity and functionality. They are more complex in terms of building and design, but they are also more versatile.

* **Technicality**

When discussing static vs. dynamic websites from a technical point of view, the differences between the two types of websites become even clearer. Because static websites only contain fixed content, building them can easily be done in plain HTML. The only way that a user can interact with a static page is by clicking hyperlinks and filling in forms (such as a contact form).

Dynamic websites are ultimately based on HTML and CSS as well, but server-side scripting is required to make them functional. HTML coding is used to create the basic design elements, while server-side languages are used to manage events and control actions that may occur on the dynamic page. For example, a WordPress website built with a theme like [Ombrello](https://templatic.com/wordpress-corporate-business-themes/agency-theme/) is a dynamic website.

* **Coding**

To create a static website, the user doesn’t need to use complex software programs. Some knowledge in HTML and CSS along with Notepad should be enough to build a simple static website. Static pages include elements such as text and multimedia elements. They are not as technical as a dynamic website, but they are not as effective either. Users will see the same design and content each time they visit the website unless you change the source code manually.

A dynamic website generates the content and displays it based on what actions the users make on the page. The preferences of the user alter what is displayed to them, which can be an intricate process based on the sophistication of the website. A special editor, such as an IDE (Integrated Development Environment), is required to build dynamic websites, along with strong technical skills in server-side language programming.

**1.4 Laravel**

**1.4.1 What is Laravel?**

Laravel is a web application framework with expressive, elegant syntax. A web framework provides a structure and starting point for creating your application, allowing you to focus on creating something amazing while we sweat the details

Laravel strives to provide an amazing developer while providing powerful features such as thorough dependency injection, an expressive database abstraction layer, queues and scheduled jobs, unit and integration testing, and more.

Whether you are new to PHP or web frameworks or have years of experience, Laravel is a framework that can grow with you. We will help you take your first steps as a web developer or give you a boost as you take your expertise to the next level. We can't wait to see what you build.

**1.4.2 Why Laravel?**

A Progressive Framework

Laravel is incredibly scalable. Thanks to the scaling-friendly nature of PHP and Laravel's built-in support for fast, distributed cache systems like Redis, horizontal scaling with Laravel is a breeze. In fact, Laravel applications have been easily scaled to handle hundreds of millions of requests per month.

#### A Community Framework

#### Laravel combines the best packages in the PHP ecosystem to offer the most robust and developer friendly framework available. In addition, thousands of talented developers from around the world have [contributed to the framework](https://github.com/laravel/framework). Who knows, maybe you will even become a Laravel contributor

## 1.4.3 [Creating Validators](https://laravel.com/docs/8.x/validation#manually-creating-validators)

If you do not want to use the validate method on the request, you may create a validator instance manually using the validator façade. The make method on the façade generates a new validator instance:Text

Description automatically generated

**1.4.4 Testing Files Uploads**

The Illuminate\Http\Uploaded File class provides a fake method which may be used to generate dummy files or images for testing. This, combined with the Storage facade's fake method, greatly simplifies the testing of file uploads. For example, you may combine these two features to easily test an avatar upload form:

Graphical user interface, text

Description automatically generated with medium confidence

PHP

PHP is a server-side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites. It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.

PHP is a recursive acronym for "PHP: Hypertext Preprocessor".

**1.4.5 Characteristics of PHP**

* Simplicity
* Efficiency
* Security
* Flexibility
* Familiarity

## "Hello World" Script in PHP

*Graphical user interface, text, application

Description automatically generated*

**1.5 Node.js**

**1.5.1 What is Node.js?**

Many developers have heard of this framework in the recent period, as this framework has garnered the internet of many programmers in the field of web development because of the somewhat mythical things it offers and a whole new level in this world.

The Node.js is a framework for programming with many modules in the JavaScript language developed by an American programmer who lives in Germany named Ryan Dahl and Super phase Joint, and the first launch of this pattern was in 2009

**1.5.2 Node.js features:**

Node.js has a large number of advantages and they are:

1- It works with asynchronous system. This word has a very clear meaning.it means asynchrony. What is meant here in programming is the ability of the server or computer to carry out more than one software processing at the same time, which makes the program, application, or site faster and more effective with the user , and this is what you give it to you.node.js .library

2- Based on the previous system in which the library operates, the language became famous for working in the non-blocking system. Here is an explanation of this system: in normal cases, when programming in most programming languages, you write the codes in the order and the computer executes the commands step by step according to the order of the code that you wrote.

3- Through node.js, you can write a JavaScript code that connects the server and the browser, so that any change or update of data that occurs in the server is executed immediately in front of the user.

4- Good handling of databases Considering that the JavaScript language can write and store simple data in json files and deal with it easily, node.js is distinguished in dealing with database that run no sql system such as mongodb and couchdh databases ,where this type of database originally depends on the concept of member in Java Script object notation language.

5- Node package manager or npm:

npm This ia a store with a very large number of tools, packages and framework that will help you build your project using node.js and they are all free

**1.5.3 Why is Node.js fast?**

Node.js's high speed is mainly due to two factors:

The advanced Chrome V8 engine and the mechanism of receiving and sending inputs and outputs I\O ,which is called non-blocking in English, as opposed to the blocking mechanism pursued by other language, especially PHP.

**1.6 C#**

In this section we discuss just how important some of C#’s functionalities are, and why it’s one of the best choices to use when it comes to windows applications.

**1.6.1 Future of C#:**

Today, C# is not only a Windows development programming language but can be used to build Web applications, Windows store apps, and mobile apps including iOS and Android. C# can also do more than that. If you’ve not already read my article, I highly recommend going and reading

**1.6.2 What C# can do for you?**  
  
At the Build 2016 event, Microsoft made several exciting announcements and one of them was integrating Xamarin as a part of Visual Studio “15” and beyond. Now C# developers can build IOS and Android apps that can spit out native IOS and Android code.  
  
...the future of C# is very bright, why?

* You can write C# in any editor you want.
* C# is open source now
* C# runs on Windows, Mac, and Linux
* C# can be used to build Windows client apps, Windows Store apps, iOS apps, and Android aps and can also be used to build backend and middle-tier frameworks and libraries.
* C# (via Roslyn, the C# engine):
* Supports all IDEs and editors
* All the linters and analysis tools
* All the fixing and refactoring and code generation tools
* All the scripting and all the REPLs
* C# 7 comes with new features including tuples, record types, and pattern matching.
* C# is young and evolving.
* Unlike other programming languages, C# is still young and evolving. Now being open sourced, C# is getting community involvement and new features are being decided by community.

**1.6.3 Abstraction in C#**

The word abstract means a concept or an idea not associated with any specific instance. In programming we apply the same meaning of abstraction by making classes not associated with any specific instance. The abstraction is done when we need to only inherit from a certain class, but do not need to instantiate objects of that class. In such case the base class can be regarded as "Incomplete". Such classes are known as an "Abstract Base Class".  
  
**1.6.4 Abstract Base Class**  
  
there are some important points about Abstract Base Class

1. An Abstract Base class can't be instantiated; it means the object of that class can't be created.
2. Class having abstract keyword and having abstract keyword with some of its methods (not all) is known as an Abstract Base Class.
3. Class having Abstract keyword and having abstract keyword with all of its methods is known as pure Abstract Base Class.
4. The method of abstract class that has no implementation is known as "operation". It can be defined as abstract void method ();
5. An abstract class holds the methods but the actual implementation of those methods is made in derived class.

**1.6.5 Encapsulation in C#**

The object oriented programming will give the impression very unnatural to a programmer with a lot of procedural programming experience. In Object Oriented programming Encapsulation is the first place. Encapsulation is the procedure of covering up of data and functions into a single unit (called class). An encapsulated object is often called an abstract data type. In this article let us see about it in a detailed manner.

**1.6.6 Need for Encapsulation**The need of encapsulation is to protect or prevent the code (data) from accidental corruption due to the silly little errors that we are all prone to make. In Object oriented programming data is treated as a critical element in the program development and data is packed closely to the functions that operate on it and protects it from accidental modification from outside functions.  
  
Encapsulation provides a way to protect data from accidental corruption. Rather than defining the data in the form of public, we can declare those fields as private. The Private data are manipulated indirectly by two ways. Let us see some example programs in C# to demonstrate Encapsulation by those two methods. The first method is using a pair of conventional accessor and mutator methods. Another one method is using a named property. Whatever be the method our aim is to use the data without any damage or change.

**1.6.7 ENCAPSULATION USING PROPERTIES**

Properties are a new language feature introduced with C#. Only a few languages support this property. Properties in C# helps in protect a field in a class by reading and writing to it. The first method itself is good but Encapsulation can be accomplished much smoother with properties.

**1.6.8 Inheritance in C#**

Inheritance is one of the three foundational principles of Object-Oriented Programming (OOP) because it allows the creation of hierarchical classifications. Using inheritance you can create a general class that defines traits common to a set of related items. This class can then be inherited by other, more specific classes, each adding those things that are unique to it.

In the language of C#, a class that is inherited is called a base class. The class that does the inheriting is called the derived class. Therefore a derived class is a specialized version of a base class. It inherits all of the variables, methods, properties, and indexers defined by the base class and adds its own unique elements.

**1.6.9 Polymorphism in C#**

Polymorphism means the same operation may behave differently on different classes.

* Example of Compile Time Polymorphism: Method Overloading
* Example of Run Time Polymorphism: Method Overriding
* Example of Compile Time Polymorphism
* Method Overloading: Method with same name but with different arguments is called method overloading.
* Method Overloading forms compile-time polymorphism.

Note - By default functions are not virtual in C# and so you need to write “virtual” explicitly. While by default in Java each function are virtual.

**1.6.10 Delegates in C#:**

Delegate is one of the base types in .NET. Delegate is a class, which is used to create delegate at runtime.  
  
Delegate in C# is similar to a function pointer in C or C++. It's a new type of object in C#. Delegate is very special type of object as earlier the entire the object we used to defined contained data but delegate just contains the details of a method.  
  
**1.6.11 Need of delegate**  
  
There might be a situation in which you want to pass methods around to other methods. For this purpose we create delegate.  
  
A delegate is a class that encapsulates a method signature. Although it can be used in any context, it often serves as the basis for the event-handling model in C# but can be used in a context removed from event handling (e.g. passing a method to a method through a delegate parameter).  
  
One good way of understanding delegates is by thinking of a delegate as something that gives a name to a method signature.

**1.6.12 Delegate magic**  
  
In class we create its object, which is instance, but in delegate when we create instance that is also referred to as delegate (means whatever you do you will get delegate).  
  
Delegate does not know or care about the class of the object that it references. Any object will do; all that matters is that the method's argument types and return type match the delegate's. This makes delegates perfectly suited for "anonymous" invocation.  
  
**Benefit of delegates**  
  
In simple words delegates are object oriented and type-safe and very secure as they ensure that the signature of the method being called is correct. Delegate helps in code optimization.  
  
**Types of delegates**

1. Singlecast delegates
2. Multiplecast delegates

Delegate is a class. Any delegate is inherited from base delegate class of .NET class library when it is declared. This can be from either of the two classes from System. Delegate or System. Multicast Delegate.

**Singlecast delegate**  
  
Singlecast delegate point to single method at a time. In this the delegate is assigned to a single method at a time. They are derived from System. Delegate class.  
  
**Multicast Delegate**  
  
When a delegate is wrapped with more than one method that is known as a multicast delegate.  
  
In C#, delegates are multicast, which means that they can point to more than one function at a time. They are derived from System. Multicast Delegate class.

**1.6.13 Collections in C#:**

“.NET” offers a variety of collections, such as ArrayList, Hashtable, queues, Dictionaries. Collections are abstractions of data algorithms. An ArrayList is an abstract dynamic array, a Hashtable collection abstracts a lookup table, a Queues collection abstracts queues and so on. In addition to that, collections implement the ICollection, IEnumerable and IClonable interfaces. The detailed specification for each collection is found under the System .Collection namespace.

**1.6.14 Exception Handling in C#:**

Exception handling is a built-in mechanism in .NET framework to detect and handle run time errors. The .NET framework contains many standard exceptions. The exceptions are anomalies that occur during the execution of a program. They can be because of user, logic or system errors. If a user (programmer) does not provide a mechanism to handle these anomalies, the .NET run time environment provides a default mechanism that terminates the program execution.  
  
C# provides the three keywords try, catch and finally to do exception handling. The try block encloses the statements that might throw an exception whereas catch handles an exception if one exists. The finally can be used for doing any clean-up process.

**1.6.15 File Handling in C#:**

The System.IO namespace provides four classes that allow you to manipulate individual files, as well as interact with a machine directory structure. The Directory and File directly extends System. Object and supports the creation, copying, moving and deletion of files using various static methods. They only contain static methods and are never instantiated. The FileInfo and DirecotryInfo types are derived from the abstract class FileSystemInfo type and they are typically, employed for obtaining the full details of a file or directory because their members tend to return strongly typed objects. They implement roughly the same public methods as a Directory and a File but they are stateful and the members of these classes are not static.

In the .NET framework, the System.IO namespace is the region of the base class libraries devoted to file based input and output services. Like any namespace, the System.IO namespace defines a set of classes, interfaces, enumerations, structures and delegates. The following table outlines the core members of this namespace.

**1.6.16 Reading and Writing to Files**  
  
Reading and writing operations are done using a File object.

**Stream**  
  
The .NET provides many objects such as FileStream, StreamReader/Writer, BinaryReader/Writer to read from and write data to a file. A stream basically represents a chunk of data flowing between a source and a destination. Stream provides a common way to interact with a sequence of bytes regardless of what kind of devices store or display the bytes.

**FileStream**A FileStream instance is used to read or write data to or from a file. In order to construct a FileStream, first we need a file that we want to access. Second, the mode that indicates how we want to open the file. Third, the access that indicates how we want to access a file. And finally, the share access that specifies whether you want exclusive access to the file.

The File Stream can read or write only a single byte or an array of bytes. You will be required to encode the System. String type into a corresponding byte array. The System. Text namespace defines a type named encoding that provides members that encode and decode strings to an array of bytes. Once encoded, the byte array is persisted to a file with the FileStream. Write() method. To read the bytes back into memory, you must reset the internal position of the stream and call the Read Byte() method. Finally, you display the raw byte array and the decoded string to the console.

**BinaryReader and BinaryWriter**  
The BinaryReader and Writer class allows you to read and write discrete data types to an underlying stream in a compact binary format. The BinaryWriter class defines a highly overloaded Write method to place a data type in the underlying stream.

**1.7 Vue.js**

**1.7.1 What is Vue.js?**

Simply it is a framework that was developed to help build interactive interfaces mainly, it is true that there are many other frameworks through Which the same thing can be done, but Vue.js is mainly built to achieve this goal, so you will get the best results by using this framework, and also In the official documentation it focuses on the concepts of user experience and improving user interaction with interface, and this is what you do not find sufficiently supported in other frameworks.

The framework supports the concept of MVVM, which is of course an abbreviation for Model-View-ViewModel, aconcept that aims to improve the interaction of the graphical interface with the back-end and here he means what is going on within the framework itself, and also the framework architecture helps to design and build singel-page applications

**1.7.2 Features of Vue.js library**

Dynamic HTML Templates Vue.Js Providers a declarative and easy-to-read method for displaying views based on HTML templates flavored with JavaScript. .....

Condition and loops Besides v-bind and v-on ,there are other Directives that enable us to control the structure and structure of the DOM tree. .....

Handling user input......

Support the idea of components.

**1.7.3 Features of Vue.js?**

**1.7.3.1 Dynamic HTML Templates:**

Vue.Js provides a declarative and easy-to-read method for displaying views based on HTML templates flavored with JavaScript. This reminds us of the first version of Angular which used the same method.

**1.7.3.2 Terms and episodes**

Besides v-bind and v-on, there are other directives that enable us to control the structure of the DOM tree.

**The conditions:**

For example, we can use v-if to display or remove a section of HTML if a certain condition is met . And if you were an old AngularJs 1.x user , you should now remember ng-if,which provides the same service.

**1.7.3.3 Handle user input**

JavaScript allows us to interact with the user by firing events, and the Vue.Js library provides a great API for taking events and binding them to the state data of our version of Vue.

**1.7.3.4 Support the idea of components:**

The concept or system of components has exploded in the web industry in the last decade with the rise of the React.js library and the idea of web components that work is in full swing to widest range of different browsers.

**1.7.4 How to Build a Desktop Application with Electron and Vue:**

Front-end Developer Wojciech works with American actor, Chuck Norris, to demonstrate how easy it is to creat a desktop app with Electron and Vue.

First, let's take a look at the process behind Electronic application.

**1.7.5 Main process**

The process that runs package.js's script is called the main process. The main process has the power of Node.js APIs And is allowed access to native resources of the system .An Electron app always has one main process

**1.7.6 Renderer process**

Electron uses Chromium's multi-process architecture to display web pages. Each web page in Electron runs its own separate process which is called are nderer process. In most cases, you will also have a single renderer process which is usually just a JavaScript application combined with your preferred JS framework

**1.7.7 Communication**

When it comes to communication between the main process and renderer process, Electron provides two IPC(Inter-process Communication) modules called ipcMain and ipcRenderer:

-The ipcMain module is used to communicate asynchronously from the main process to the renderer process.

-The ipcRenderer module communicates asynchronously from the render process to the main process.

**1.8 Flutter**

**1.8.1 What is Flutter?**

Flutter is a free and open-source mobile UI framework created by Google and released in May 2017. In a few words, it allows you to create a native mobile application with only one codebase. This means that you can use one programming language and one codebase to create two different apps (for IOS and Android).

Flutter consists of two important parts:

* An SDK (Software Development Kit): A collection of tools that are going to help you develop your applications. This includes tools to compile your code into native machine code (code for IOS and Android).
* A Framework (UI Library based on widgets): A collection of reusable UI elements (buttons, text inputs, sliders, and so on) that you can personalize for your own needs.

**1.8.2 Why Flutter?**

**1. Fast Development**

Flutter is faster than many other application development frameworks. With its “**hot reload**” feature, you can experiment, build UIs, add/remove features, test and fix bugs faster. Thus reducing the overall app development time.

2. **Expressive and Flexible UI**

You can really build beautiful apps in Flutter. Also, the end-user experience is similar to native apps. Flutter has a layered architecture that lets you control every pixel on the screen. Thus, customization is very simple in Flutter. With its powerful composting capabilities, you can overlay and animate graphics, text, video, and other controls without any limitations.

**3. Native Performance**

Flutter’s widgets incorporate all critical platform differences such as scrolling, navigation, icons and fonts. This provides a native performance experience on both IOS and Android.

4. **Dart Language**

Dart programming language is developed by Google and is meant for mobile, desktop, backend and web applications. It is a client-optimized language for fast performing apps on multiple platforms.  
Dart is AOT (Ahead Of Time) compiled to fast, predictable, native code, allowing writing almost all of Flutter code in Dart. This makes Flutter extremely fast and customizable. Virtually, everything (including all the widgets) can be customized.

### ****5. Important Flutter Tools****

Flutter framework supports many different tools including Android Studio and Visual Studio Code. It also provides support for building apps from the command line. **Dart DevTools**, which is a new debugging tool, is more flexible and allows runtime inspection. You can also view logs, debug apps and inspect widgets for Flutter App Development.

1. **Widget inspector**helps to visualize and explore the tree hierarchy. Flutter uses this for UI rendering.
2. **Timeline view**helps you to monitor your application at a frame-by-frame level. You can also identify rendering and computational work in timeline view.
3. **Source-level Debugger:** It lets you step through code, set breakpoints and investigate the call stack.
4. **Logging View**displays events from the Dart runtime, application frameworks and app-level logging events.

**1.8.3 What is Flutter feature?**

**1. Same UI and Business Logic in All Platforms**

**2. Reduced Code Development Time**

**3. Increased Time-to-Market Speed**

**4. Similar to Native App Performance**

**5. Custom, Animated UI of Any Complexity Available**

**6. Own Rendering Engine**

**7. Simple Platform-Specific Logic Implementation**

**8. The Potential Ability to Go Beyond Mobile**

### 1.8.4 Flutter VS React native

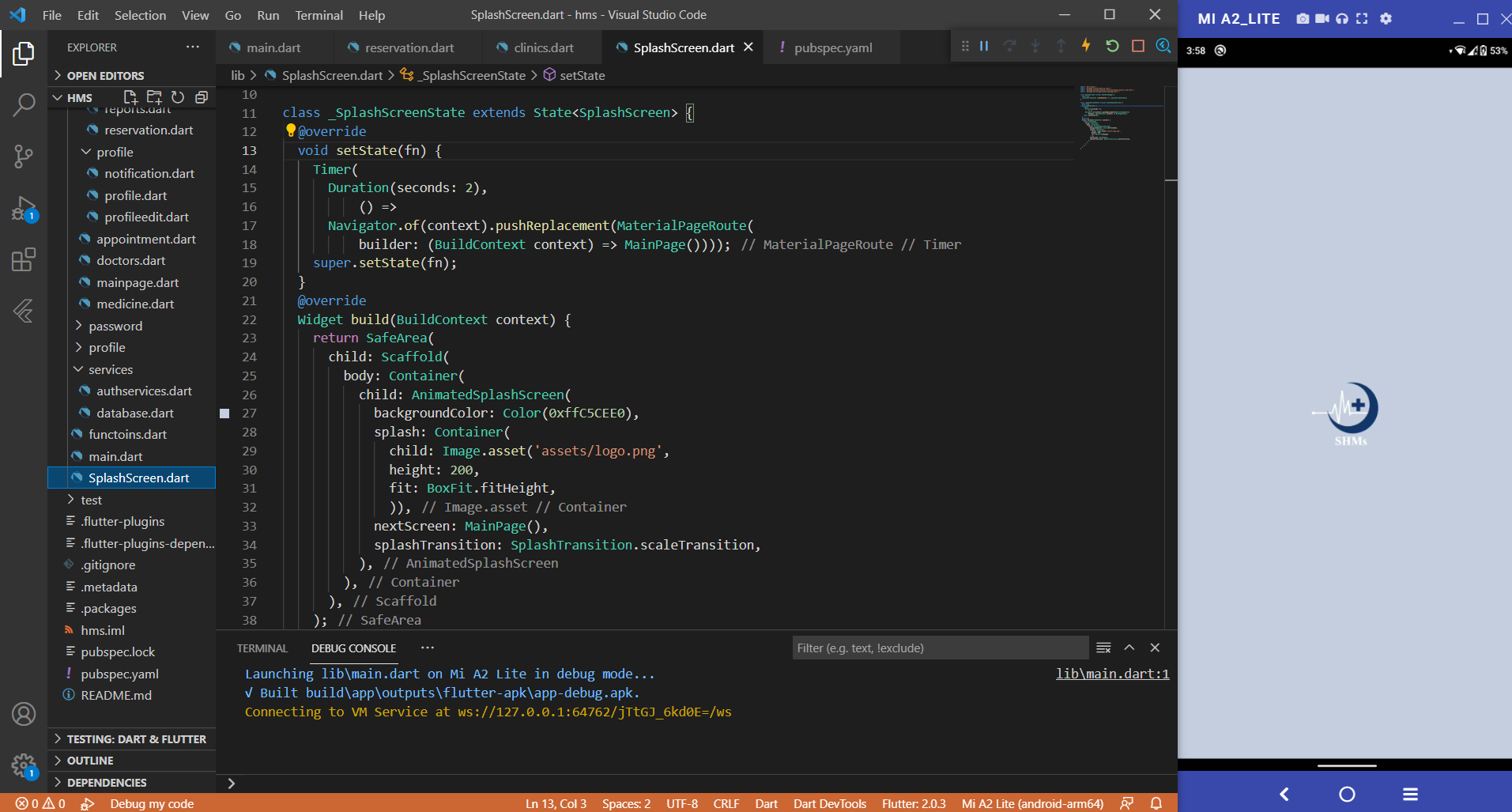
### Screenshot (86)

**1.8.5 Library we use in the Project**

**1.8.1 Animated Splash Screen**

We use it to make welcome page

Dependencies: animated\_splash\_screen: ^1.1.0



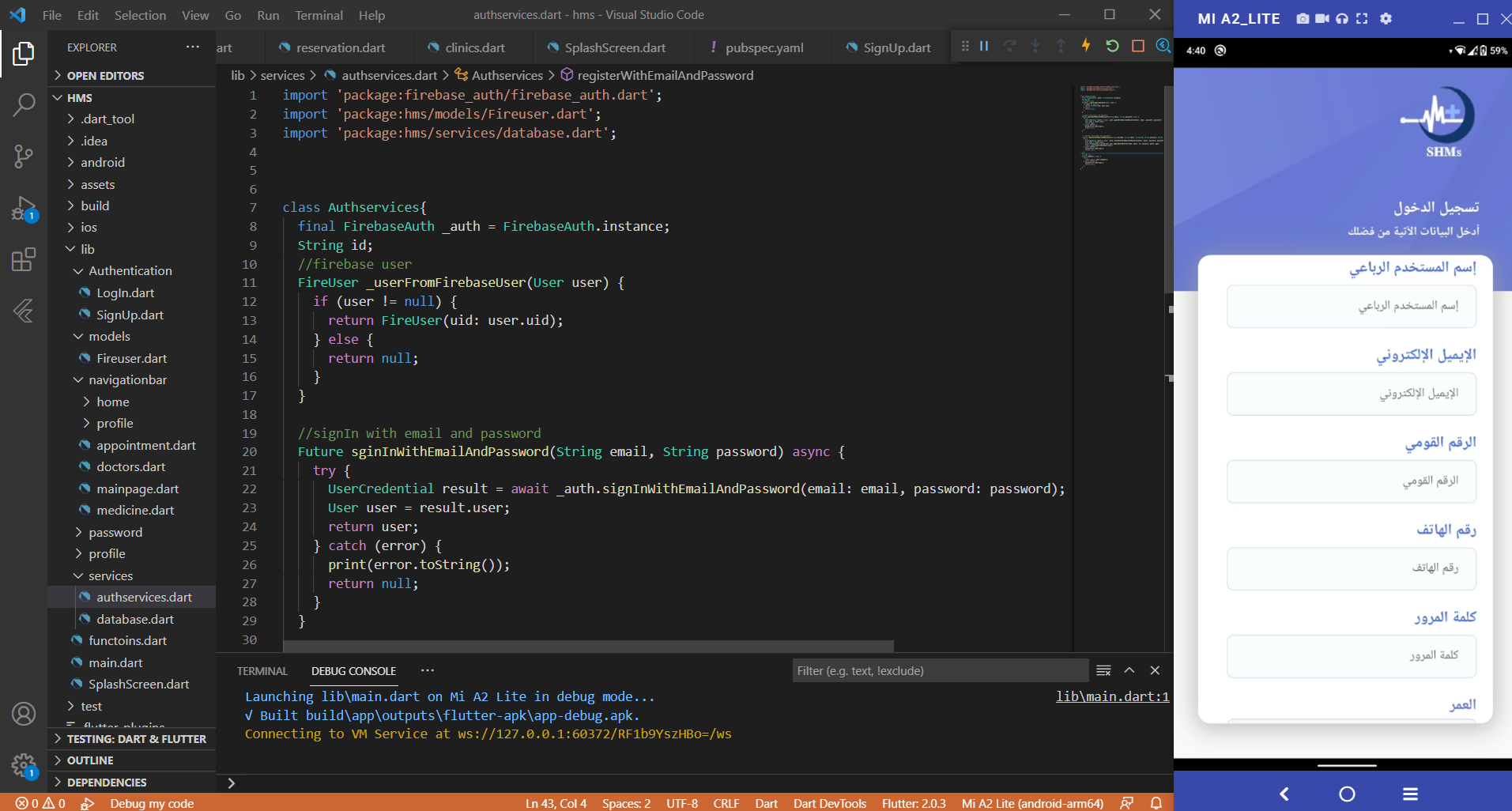
**1.8.2 Firebase Authentication**

## Authentication state

Firebase Authentication provides many methods and utilities for enabling you to integrate secure authentication into your new or existing Flutter application. In many cases, you will need to know about the authentication state of your user, such as whether they're logged in or logged out.

We use it in the Project to authentication the sign in and register an account

dependencies: firebase\_auth: ^1.4.1

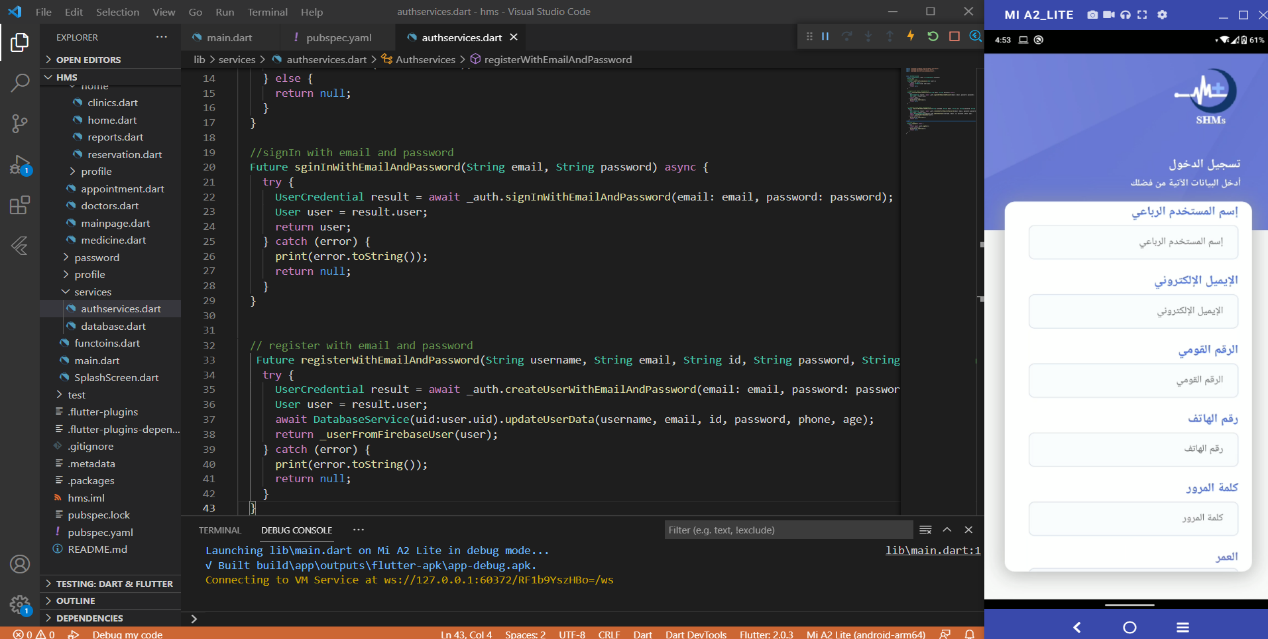


**1.8.3 Cloud Firestore**

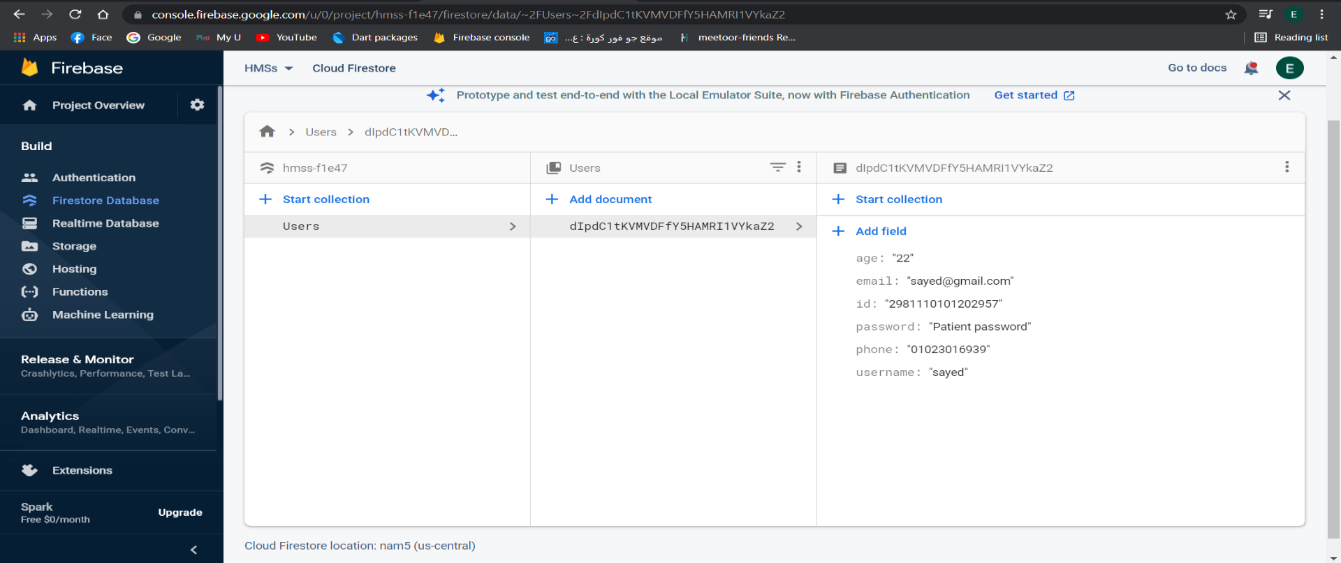
## Collections & Documents

## Firestore stores data within "documents", which are contained within "collections". Documents can also contain nested collections. For example, our users would each have their own "document" stored inside the "Users" collection. The collection method allows us to reference a collection within our code.

dependencies: cloud\_firestore: ^2.2.2



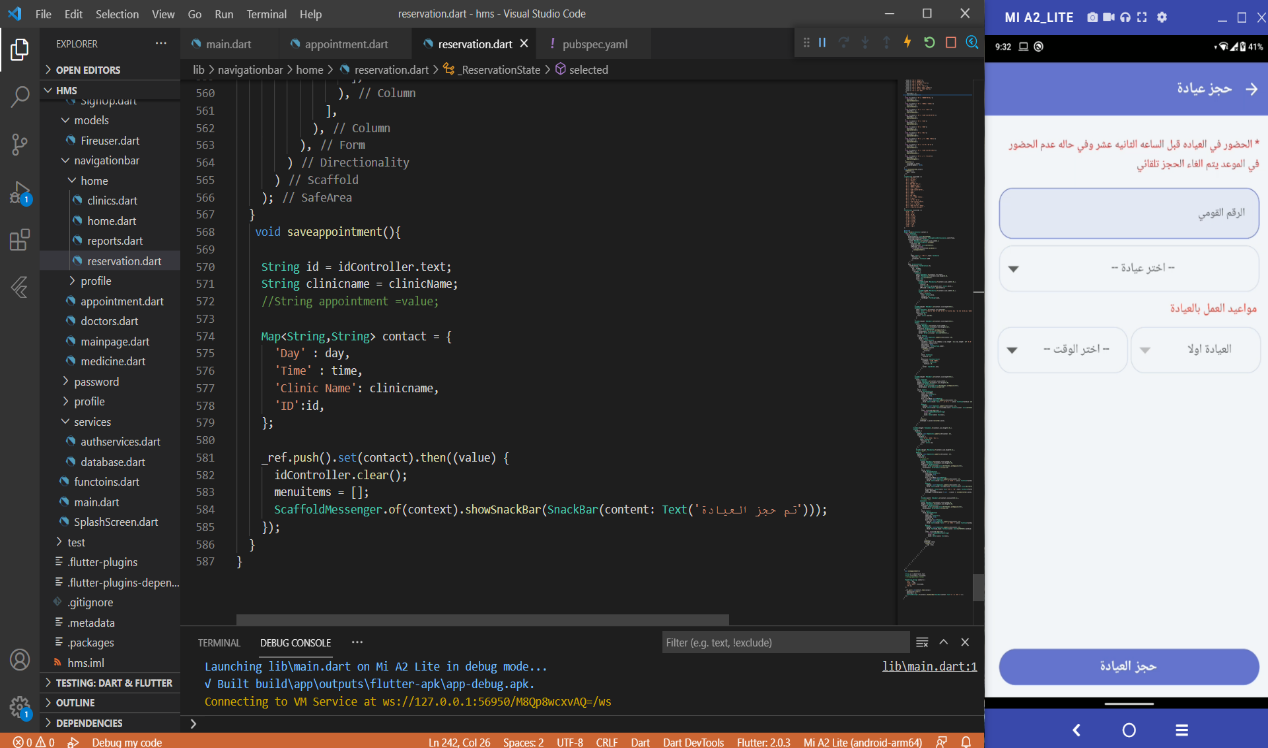
here we use cloud\_firestore to store the input the user enter and there is an example for the registration output in firebase console



* + 1. **Firebase database**

We use this library in the project to store the reservation inputs into database to show them into an appointment page

dependencies: firebase\_database: ^7.1.1



All of inputs ID, clinic name, and appointment store in firebase database

And the output

