**B NEW MOBILES**

**A Mini-Project Report**

Submitted for the partial fulfillment for the award of degree of

**BACHELOR OF SCIENCE IN COPUTER SCIENCE**

Submitted by

KAVIYA.R(A17CSEB13)

MAHALAKSHMI.P(A17CSEB14)

Under the Guidance of

**MR.A.ARUN BENEDICT M.Sc.,M.Phil.,**

Assistant Professor

(Post Graduate and Research Department of Computer science)



**ST.JOSEPH'S COLLEGE OF ARTS & SCIENCE (AUTONOMOUS)**

(Affiliated to Thiruvalluvar University-Vellore)

CUDDALORE-607001

**May-2020**

**CERTIFICATE**

This is to certify that the Mini-Project entitled

**B NEW MOBILES**

Being submitted to

**ST.JOSEPH'S COLLEGE OF ARTS & SCIENCE (AUTONOMOUS)**

(Affiliated to Thiruvalluvar University-Vellore)

Submitted by

R.KAVIYA(A17CSEB13)

P.MAHALAKSHMI(A17CSEB14)

For the Partial Fulfillment For the award of degree of

**BACHELOR OF SCIENCE IN COMPUTER SCIENCE**

Is a Bonafide record of work carried out by them,

under my guidance and supervision.

INTERNAL GUIDE HEAD OF THE DEPARTMENT

Submitted for the viva -voce examination held on..............................................................

EXAMINERS:

1. ……..
2. ……..

**ACKNOWLEDGEMENT**

It is our earnest and sincere desire and ambition to acquire profound knowledge in the study of Bachelor in Computer Science . I am grateful to God, the Almighty who has blessed me abundantly and guided me to complete this task

We express our sincere thanks to most respectful Secretary,**Rev.Fr.G.PETER RAJENDIRAM,M.A.,M.Sc.,M.Ed.,M.Phil.,** St.Joseph's College of Arts and Science (Autonomous)Cuddalore,for providing such a good congenial environment to enlighten my Knowledge.

I am happy to express my heartfelt thanks to Principal and Head of the Department **Dr.M.ARUMAI SELVAM,M.Sc.(Maths), M.Sc.(CS), M.Phil.(CS), PGDCA,Ph.D.,** for the motivation and encouragement in successful completion of the project.

I take immense pleasure in conveying my sincere and deepest gratitude to my internal guide **MR.A.ARUN BENEDICT M.Sc.,M.Phil.,** Assistant Professor, Post Graduate and Research Department of Computer Science for his valuable guidance ,inspiration, consent and motivation throughout the development of our project.

At the outset I extend my sincere gratitude to all our Staff of Computer Science for having rendered all Support whenever needed.

I will fail in my duty if I do not put on record my heartfelt gratitude to my beloved parents for their financial and material support as well as their words of encouragement and motivation to complete this project successfully.

Last but not Least , I place my deepest sense of gratitude to all our Friends who had inspired and supported me to complete this project more successfully.

R.KAVIYA P.MAHALAKSHMI

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| S.NO | CONTENT | PAGE.NO |
| 1. | **ABSTRACT**  **INTRODUCTION**  1.1 About the project | 1  2 |
| 2. | **SYSTEM ANALYSIS**  2.1 PROBLEM DESCRIPTION  2.1.1 Existing System  2.1.2 Proposed System  2.2 FEASIBILITY STUDY   * + 1. Economic Feasibility     2. Operational Feasibility   2.2.3 Technical Feasibility  2.3 SYSTEM SPECIFICATION  2.3.1 Hardware requirements  2.3.2 Software requirements | 3  4  5 |
| 3. | **SYSTEM DESIGN**  3.1 MODULE DESCRIPTION     * + 1. Home     2. About us   3.1.3 User Details  3.1.4 Mobiles  3.1.5 Feedback  3.2 REQUIREMENTS ANALYSIS  3.3 SYSTEM DESIGN  3.3.1 Admin process  3.3.2 User process  3.4 DATABASE DESIGN | 6  7  8  9 |
| 4. | **SYSTEM TESTING**  4.1 SYSTEM TESTING  4.1.1 Unit testing  4.1.2 Integration testing  4.1.3 Alpha testing | 10    12 |
| 5. | **CONCLUSION AND FUTURE ENHANCEMENT**  5.1 CONCLUSION  5.2 FUTURE ENHANCEMENT | 13 |
| 6.  7. | **REFERENCES**  6.1 WEBSITES    6.2 BOOKS  **APPENDIX**  APPENDIX-A  SOFTWARE DESCRIPTION  APPENDIX-B  SAMPLE SOURCE CODE  APPENDIX-C  SCREENS | 14  15 |

**ABSTRACT**

This project entitled **“B NEW MOBILES”** a web based project with an e-commerce platform for purchase of mobile through online. The user can easily purchase online by simple procedure like login, add to cart, then purchase the mobile of their choice and payment is done very securely through online payment process. User login, with their unique User id and Password. Every new user has to register to purchase through this website. After login the website has a Gallery, Registration form, About us, Blogs etc.

**CHAPTER-1**

**INTRODUCTION**

* 1. **ABOUT THE PROJECT:**

Now a days the life style of the people is different and everything has been processed in online. People feel uncomfortable and time consuming for going crowded markets. So, E-Shopping is a boon as it saves lot of time. Online shopping is a process whereby consumers directly buy goods, services etc. from a seller without an intermediary service over the Internet. Shoppers can visit web stores from the comfort of their house and shop as by sitting in front of the computer. Online stores are usually available 24 hours a day and many consumers have internet access both at work and at home. So it is very convenient for them to shop Online. One of the most enticing factors about online shopping, particularly during holiday season is, it alleviates the need to wait in long lines or search from a store for a particular item. Varieties of goods are available in online. So the researcher wants to know the preference of the consumers. So fifty respondents were met and data were collected regarding their preference towards shopping online.

**CHAPTER-2**

**SYSTEM ANALYSIS**

**2.1 PROBLEM DESCRIPTION**

**2.1.1 EXISTING SYSTEM:**

Mostly in the existing system most of the e-commerce website has been developed through many programming language like HTML, JAVA etc. in the front end as well as backend with the database that has a lot of work. And the existing site has a website with more complex usability that most of people find difficult in understanding & using in a day to day life.

**2.1.2 PROPOSED SYSTEM:**

The ultimate goal of this project is for an easy usability of a website for all common peoples. It has simple page navigation, form page, displays only branded mobiles for a affordable price. This project is developed using ASP.NET as fronted and GOOGLE SUITE as backend and GOOGLE SHEETS.

**2.2 FEASIBILITY STUDY**

A Feasibility study is a preliminary study undertaken to determine and document a project’s viability. The results of this study are used to make a decision whether to proceed with the project, or table it. If it indeed leads to a project being approved, it will-be-before the real work of the proposed project starts-be used to ascertain the likelihood of the project’s success .It is an analysis of possible alternative solutions to a problem and a recommendation on the best alternative. It , for example, can decide whether an order processing be carried out by a new system more efficiently than the previous one .there are different aspects in the feasibility study

* Economic feasibility study
* Technical feasibility study
* Organization feasibility study

**2.2.1 ECONOMIC FEASIBILITY:**

* The computerized system takes care of the present existing system data flow and procedures completely and should generate all the reports of the manual system besides a host of other management reports
* It should be built as a web based application with a separate web server. This is required as the activities are spread throughout the organization and the customer wants a centralized database .further some of the linked transactions take place in different locations .
* Open source software like ASP.NET AND GOOGLE SUITE is used to minimize the cost for the customer

**2.2.1 TECHNICAL FEASIBILITY :**

The technical issue usually raised during the feasibility stage of the investigation includes the following

* Does the necessary technology exist to do what is suggested?
* Does the proposed equipment have the technical capacity to hold the data required to use the new system?
* Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
* Can the system be upgraded if developed?

**2.2.3 OPERATIONAL FEASIBILITY:**

This system has enough support from organizations to be implemented successfully . As the strategy is well applied on the both economic and schedule issues, it did not bring any excessive amount of change.

**2.3 SYSTEM SPECIFICATION:**

**Hardware Requirement:**

**Processor :** intel i3 7th generation

**Speed :** 2.00 GHZ

**RAM :** 4.00 GB

**System type :** 64-bit operating system

**Hard disk :** 1 TB

**Software requirements:**

**Operating system :** windows 10

**Front end :** ASP.NET

**Back end :** GOOGLE SUITE

**Editor :** Visual studio

**User interface design :** HTML, CSS, JAVASCRIPT,

**Web browser :** Google chrome

**CHAPTER-3**

**SYSTEM DESIGN**

**3.1 MODULES**

3.1.1 HOME

3.1.2 ABOUT US

3.1.3 USER DETAILS

3.1.4 MOBILES

3.1.5 FEEDBACK

**3.1.1 HOME**

The Home page has a header, footer and a body which displays the necessary content of a website. The header section has a register, banner, blog, add to cart. The footer has disclaimer content and the copy rights. This home module is used to display the details about the website and it leads the users to use the catalog page that has mobile details and shows the facilities provided by the website.

**3.1.2 ABOUT US**

This module contains the detailed information about the website.

**3.1.3 USER DETAILS**

This module contains the detailed information about the user like contact number, name, address, alternative address, alternative contact number and also the history of a orders placed.

**3.1.4 MOBILES**

This product module is a catalog page which is used to display the items of a mobile that are sold online with its product details. It offers the online purchase by the order button and its added in a cart page. Once the user want to purchase he has to click “Buy” option and now it leads to the payment page where cash on delivery only available and the user can order the product and they can buy it through this page.

**3.1.5 FEEDBACK**

In this module, the users can enter the feedbacks like rating, reviews of the website as well as the products.

* 1. **REQUIREMENT ANALYSIS:**

The requirement analysis is to create a simple website for selling of branded mobile through online which is through cash on delivery. The website is very simple, easy usability for a common people who are not aware of online purchase and all the details are stored by using Google Docs.

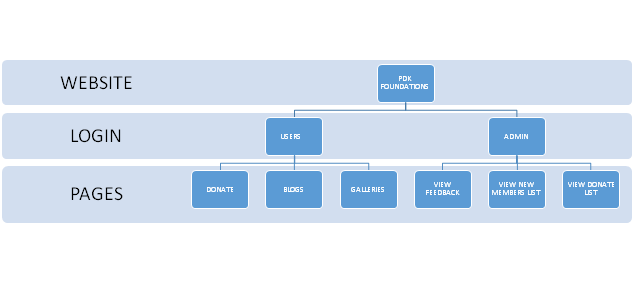
**3.3 SYSTEM DESIGN:**

System design describes the behavior of the target system from an external point of view. System design describes “the meat” of the actual requirements.

**3.3.1 ADMIN PROCESS:**

The admin process is a backend process which is done by the admin and maintains all the necessary details. The admin has rights to add, update, delete. The entire database can also be extracted by the admin for further use.

**3.3.2 USER PROCESS:**





**3.3 DATABASE DESIGN:**

**FORMS NAME.1.**

|  |  |
| --- | --- |
| **FIELD NAME** | **TYPES** |
| First name | Text type |
| Last name | Text type |
| User name | Text type |
| Phone no | Number |
| Address | Paragraph type |

**FORMS NAME.2. B NEW MOBILES (FEEDBACK)**

|  |  |
| --- | --- |
| **FIELD NAME** | **TYPES** |
| User name | Text type |
| Phone no | Text type |
| Feedback | Dropdown |
| Comment | Paragraph |

**CHAPTER-4**

**SYSTEM TESTING**

For any software that is newly developed, primary importance has been given to testing of the system. It is the responsibility of the developer to detect all possible errors in the software before handling it to the user or Customer. Since it is an Open source website it needs intensive testing to provide user friendly infrastructure.

**TESTING:**

Testing is the process by which a developer will generate a set of test data, which gives maximum probability of finding all types of errors that can occur in the software. Testing is an important phase in the software development cycle of any system design. Testing indicates the futility of attempting to detect all errors through testing alone. Exhaustive testing of all paths is seldom feasible, and even if it is feasible it does not guarantee detection of missing path errors, computational errors or domain errors. These considerations re-emphasize the need for systematic analysis and design for continuous verification of work products, so that errors are removed prior to implementation.

Apart from Black box testing, which tests the behavioral aspects of the system and white box testing, which tests the functional aspects of the system, there are various testing strategies to carry out software testing at the successful level. They are explained below as follows:

**4.1.1 UNIT TESTING:**

Unit testing or First-Level Testing comprises the set of tests performed by an individual programmer prior to integration of the Modules or units into a larger system. The system modules login, administrator, hall-manager, customer modules are being tested.

**4.1.2 INTEGRATION TESTING:**

Integration Testing is the systematic technique for constructing the program structure while conducting tests to uncover errors associated with interfacing. There are two approaches for Integration testing. The approach followed in this project is the bottom down approach as it follows object oriented language structure.

**TOP-DOWN INTEGRATION:**

It is an incremental approach to construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main module. Modules subordinate to the main module control module are incorporated into the structure in either a depth-first or breadth first manner.

**4.1.3 ALPHA TESTING:**

The Alpha testing is conducted at the developer’s site by the customer. As per their need the required modifications are also made successfully**.**

**CHAPTER-5**

**CONCLUSION AND FUTURE ENHANCEMENT**

**5.1 CONCLUSION:**

By using a simple methodology in front end and back end a simple website has be created for selling on branded mobile through online by a easy payment which is COD.

**5.2 FUTURE ENHANCEMENTS:**

The future enhancement is that the sample e-commerce website can be developed using advanced technology and options for user friendly with the payment options also.

**CHAPTER-6**

**BIBLIOGRAPHY**

**6.1 REFERENCE BOOKS**

* “Core C# and .NET – Stephen C. Perry – Pearson Education Inc. and Dorling KindersleyPublishingInc.–2006Edition.
* PROGRAMMING MICROSOFT VISUAL BASIC.NET by Alfred C.Thomson distributed by Mainfunction.com
* Asp.Net :The complete reference 1st Edition (English,Softcover,Matthew MacDonald)
* C# in depth- Jon Skeet - Manning publicationspap/pas edition(April 21 2008)

**6.2 REFERENCE WEBSITES**

* [**www.w3schools.com**](http://www.w3schools.com/)

* [**www.codepen.com**](http://www.4shared.com/)

* [**www.dynamicdrive.com**](http://www.dynamicdrive.com/)

**CHAPTER - 7**

**APPENDIX A**

**SOFTWARE DESCRIPTIONS**

**MICROSOFT VISUAL STUDIO**

Microsoft visual studio is an integrated development environment (IDE) from Microsoft. It can be used to develop console and graphical user interface applications along with Windows forms applications, Web sites, Web applications, and Web services in both native code together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework,

.NET compact Framework and Microsoft Silverlight .

Visual studio includes a code editor supporting Intelligence as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built in tools include a forms designer for building GUI applications, Web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level-including adding support for source-control systems (like Subversion and Visual SourceSafe) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development life cycle (like the Team Foundation Server client: Team Explore).

Visual Studio supports different programming languages by means of language services, which allow the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C/C++ (Via Visual C++), VB.NET (Via Visual Basic

.NET), C# (Via Visual C#), and F#(as of Visual studio 2010[2]). Support for other languages such as M, python, and Ruby among others is available via language services installed separately. It also supports XML/XSLT, HTML/XHTML, Java script and CSS. Individual versions of Visual Studio also exit which provide more limited language services to the user: Microsoft Visual Basic, Visual J#, Visual C#, and Visual C++.

Microsoft provides “Express” editors of its Visual Studio 2010 components Visual Basic, Visual C#, Visual C++, and Visual Web Developer at no cost. Visual Studio 2010, 2008 and 2005 professional Editors, along with language-specific versions (visual Basic, C++, C#, J#) of Visual Studio 2005 are available for free to students as downloads via Microsoft’s DreamSpark Program

**MICROSOFT.NET**

Microsoft.Net is one of the latest and new technologies introduced by Microsoft Corporation. Nowadays we use to connect to the internet using a computer and a remote computer response via a web page and a collection of Web pages are called Web sites. The concept in .NET is that these Websites can integrate with other sites and services using standard protocols like HTTP.

Microsoft.NET platform comprises of four core components such as:

NET Building Block Services such as file storage, calendar called passport.NET

NET Device software which will run on latest Internet Devices like Mobile Phones.

NET user experience such as integrating this technology to user created documents (integrates with XML). For example if you code XML via a .NET Language like C#, it will automatically create XML documents.

Net infrastructure which includes NET Framework (Common Language Runtime & .NET Framework Class Libraries) Microsoft Visual Studio.NET such as Visual Basic.NET, Visual C++.NET etc. NET Enterprise Servers and Microsoft Windows.NET.

We can robust, Scalable, distributed applications with the help of .NET and the part that helps us to develop these applications is called the >NET Framework. The .NET Framework contains Common Language Runtime (CLR) and the .NET Framework class libraries.

All the .NET languages (like C-sharp, Visual Basic.NET, Visual C++.NET etc.) have the .NET Framework class libraries built into them. The .NET class also supports File I/O, database operations, EML (Extensible Mark-up Language) and SOAP (Simple Object Access Protocol). For example you can develop Xml pages by using C-sharp language.

**TERMS ASSOCIATED WITH MICROSOFT.NET LANGUAGE RUNTIME**

Common Language Runtime also called CLR provides a universal execution engine for developer’s code. It generates SOAP when it makes remote procedure calls. CLR is independent and is provided as part of the .NET Framework. The main features of CLR are as follows:

Manages Code

Automatic application installation

Memory management

Automatic Garbage Collection

Very high level of security while executing

.NET FRAMEWORK CLASS LIBRARIES

These class libraries works with any language under the common language runtime environment. It includes Visual Studio.NET, C-sharp. Therefore if you are familiar with one .NET language then you can easily migrate to other .NET Languages. The figure given below shows the .NET Framework hierarchy.

COMMON LANGUAGE SPECIFICATION (CLS)

It is a set of rules that a language compiler must adhere to in order to create .NET applications that run in the CLR. If you are going to create a compiler for .NET, you have to adhere to the rules enumerated in the common language specification and this enables us to create a club of CLS compliant languages

Each such compiler will have the following features:

Complete access to .NET Framework hierarchy

High level of interoperability with other compliant languages like Visual Basic

.NET etc.

**ASP.NET**

ASP.NET is a Web application framework developed and marketed NY Microsoft to allow programmers to build dynamic Web sites, Web applications and web services. It was first released in January 2002 with version 1.0 of the

.NET Framework, and is the successor to Microsoft’s Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language. The ASP.NET SOAP extension allows ASP.NET components to process SOAP messages.

**CHARACTERISTICS OF ASP.NET**

ASP.NET Web pages or Web Page, known officially as “Web forms”, are the admin building block for application development. Web forms are contained in files with an “.aspx” extension; these files typically contain static(X) HTML mark- up, as well as mark-up defining server –side Web controls and user controls where the developers place all the required static and dynamic content for the web page. Additionally, dynamic code which runs on the server can be placed in a page within a block < % -- dynamic code-- % >, which is similar to other web development technologies such as PHP, JSP, and ASP. With ASP.NET framework 2.0, Microsoft introduced a new code-behind model which allows static text to remain on the aspx page, while dynamic code remains in an aspx. Vb or aspx.cs files (depending on the programming language used).

**CODE-BEHIND MODEL**

Microsoft recommends dealing with dynamic program code by using the code

behind model, which place this code in a separate file or in a specially designated script tag. Code- behind files typically have names likes Mypage.aspx.cs or My page.aspx.vb while the page file is Mypage.aspx (same filename as the page file (ASPX), but with the final extension denoting the page language). This practice is automatic in Microsoft Visual Studio and other IDEs. When using this style of programming, the developer writes code to respond to

different events, like the page being loaded, or a control being clicked, rather than a procedural walk through of the document.

ASP.NET’s code behind model marks a departure from classic ASP in that it encourages developers to build application with separation of presentation and content in mind. In theory, this would allow a web designer, for example, to focus on the design mark-up with less potential for disturbing the programming code that drives it. This is similar to the separation of the controller from the view in Model-view-controller (MVC) frameworks.

**DIRECTIVES**

A directive is a special instruction on how ASP.NET should process the page. The most common directives is <%@ page %> which can specify many things, such as which programming language is used for the server-side code.

**USER CONTROLS**

User controls are encapsulation of sections of pages which are registered and used as controls in ASP.NET. Users controls are created as ASCX mark-up files these files usually contain the static(x) HTML mark-up, as well as mark-up defining server-side web controls. These are the locations where the developer can place the required static and dynamic content. A user control is compiled when its containing page is requested and is stored in memory for subsequent requests. User controls have their own events which are handled during the life of ASP.NET requests. An event bubbling mechanism provides the ability to pass an event fired by a user control up to its containing page. Unlike an ASP.NET page, a user control cannot be requested independently; one of its containing pages is requested instead.

**CUSTOM CONTROLS**

Programmers can also build custom controls for ASP.NET applications. Unlike user controls, these controls don’t have an ASCX mark up file, having all their code compiled into a dynamic link library (DLL) file. Such custom controls can be used across multiple web applications and visual studio projects.

**RENDERING TECHNIQUE**

ASP.NET uses a visited composites rendering technique. During compilation, the template (.aspx) file is compiled into initialization code which builds a control tree (the composite) representing the original template. Literal text goes into instances of the Literal text goes into the Literal control class, and server controls are represented by instances of a specific control class. The initialization code is combined with user-written code (usually by the assembly of multiple partial classes) and results in a class specific for the page. The page doubles as the root of the control tree.

Actual requests for the page are processed through a number of steps. First, during the initialization steps, an instance of the page class is created and the initialization code is executed. This produces the initial control tree which is now typically manipulated by the methods of the page in the following steps. As each node in the tree is a control represented as an instance of a class, the code may change the tree structure as well as manipulate the properties/methods of the individual nodes. Finally, during the rendering step a visitor is used to visit every node in the tree, asking each node to render itself using the methods of the visitor. The resulting HTML output is sent to the client.

After the requested has been processed, the instance of the page class is discarded and with it the entire control tree. This is a source of confusion among novice ASP.NET programmers who rely on class instance members that are lost with every page request/response cycle.

**STATE MANAGEMENT**

ASP.NET applications are hosted by a Web server and are accessed using the stateless HTTP protocol. As such, if an application uses stateful interaction, it has to implement state management on its own. ASP.NET provides various functions for state management. Conceptually, Microsoft treats “state” as GUI state. Problems may arise if an application needs to keep track of “data state”; for example, a finite state machines which may be in a transient state between request (lazy evaluation) or which takes a long time to initialize. State management in ASP.NET pages with authentication can make Webscraping

difficult or impossible.

**APPLICATION STATE**

Application state is held by a collection of shared user-defined variables. These are set and initialized when the Application\_Onstart event fires on the loading of the first instance of the application and are available until the last instance exits. Application state variables are accessed using the Applications collection, which provides a wrapper for the application state variables. Application state variables are identified by name. [11]

**SESSION STATE**

Server-side session state is held by a collection of user-defined session variables that are persistent during a user session. These variables, accessed using the Session collection, are unique to each session instance. The variables can be set to be automatically destroyed after a defined time of inactivity even if the session does not end. Client-side user session is maintained by either a cookie or by encoding the session ID in the URL itself.

**PERFORMANCE OF ASP.NET**

ASP.NET aims for performance benefits over other script-based technologies (including Classic ASP) by compiling the server-side code to one or more DLL files on the web server. This compilation happens automatically the first time a page is requested (Which means the developer need not perform a separate compilation step for pages). This feature provides the ease of development offered by scripting languages with the performance benefits of a compiled binary. However, the compilation might cause a noticeable but short delay to the Web user when the newly-edited page is first requested from the web server, but won’t again unless the page requested is updated further.

The ASPX and other resource files are placed in a virtual host on an Internet Information Services server (or other compatible ASP.NET servers). The first time a client requests a page, the .NET Framework parse and compiles the file(s) into a.NET assembly and sends the response; subsequent requests are served from the DLL files. By default ASP.NET will compile the entire site in batches of 1000 files upon first request. If the compilation delay is causing problems, the batch size or the compilation strategy may be tweaked.

Developers can also choose to pre-compile their “code behind” files before deployment, using MS Visual Studio, eliminating the need for just-in-time compilation in a production environment. This also eliminates the need of having the source code on the web server.

**C#.NET**

C# is a multi-paradigm programming language encompassing imperative, declarative, functional, generic, object-oriented (class-bases), and component- oriented programming disciplines. It was developed by Microsoft Within the .NET initiative and later approved as a standard by ECMA (ECMA-334) and ISO (ISO/IEC 23270). C# is one of the programming languages designed for the Common Language Infrastructure.

C# is intended to be a simple, modern, general-purpose, object-oriented programming language. Its development team is led by Anders Hejlsberg. The most recent version is C# 4.0, which was released on April 12, 2010.

Design goals

The ECMA standard lists these design goals for C#:

C# language is intended to be a simple, modern, general-purpose, object

–oriented programming language.

The language, and implementations thereof, should provide support for software engineering principles such as strong type checking, array bounds checking, detection of attempts to use uninitialized variables, and automatic garbage collection. Software robustness, durability and programmer productivity are important.

The language is intended for use in developing software components suitable for deployment in distributed environments.

Source code portability is very important, as is programmer portability, especially for those programmers already familiar with c and c++.

Support for internationalization is very important.

C# is intended to be suitable for writing applications for both hosted and embedded systems, ranging from the very large that use sophisticated operating systems, down to the very small having dedicated functions.

Although C# applications are intended to be economical with regard to memory and processing power requirements, the language was not intended to compete directly on performance and size with c or assembly language.

**FEATURES OF C# .NET**

By design, C# is the programming language that most directly reflects the underlying Common Language Infrastructure (CLI). Most of its intrinsic types correspond to the value-types implemented by the CLI framework. However, the language specification does not state the code generation requirements of the compiler; that it does no state that a C# compiler must target a Common Language Runtime, or generate Common Intermediate Language (CLI), or generate any other specific format. Theoretically, a C# compiler could generate machine code like traditional compilers of C++ or FORTRAN.

Some notable distinguishable features of C# are:

There are no global variables or functions. All methods and members must be declared within classes. Static members of public classes can substitute for global variables and functions.

Local variables cannot shadow variables of the enclosing block, unlike c and C++. Variable shadowing is often considered confusing by C++ texts.

C# supports a strict Boolean data type, bool. Statements that take conditions, such as while and if, require and expression of a type that implements the true operator, such as the Boolean type.

In C#, memory address pointers can only be used within blocks

specifically marked as unsafe, and programs with unsafe code need appropriate permissions to run. Most object access is done through safe object references, which always either point to a “live” object or have the Well –defined null value; it is impossible to obtain a reference to a “dead” object(one which has been garbage collected ), or to a random block of memory. An unsafe pointer can point to an instance of a value-type, array, string or a block of memory allocated on a stack. Code that is not marked as unsafe can still store and manipulate pointers through the system. IntPtr type, but it cannot dereference them.

Managed memory cannot be explicitly freed; instead, it is automatically garbage collected. Garbage collection addresses the problem of memory leaks by freeing the programmer of responsibility for releasing memory which is no longer needed.

In addition at the try…catch construct to handle exceptions, C# has a try…finally construct to guarantee execution of the code in the finally block.

Multiple inheritances are not supported, although a class can implement any number of interfaces. This was a design decision by the language’s lead architect to avoid complication and simplify architectural requirements throughout CLI.

C# is more type safe than C++. The only implicit conversions by default are those which are considered safe, such as widening of integers. This is enforced at compile-time, during JIT and, in some cases, at runtime. There are no implicit conversions between Boolean and integers, no between enumeration members and integers (except for literal 0, which can be implicitly converted to any enumerated type). Any user-defined conversion must be explicitly marked as explicit or implicit, unlike C++ copy constructions and conversion operators, which are both implicit by default. Starting with version 4C# supports a “dynamic” data type that enforces type checking at runtime only.

Enumeration members are placed in their own scope.

C# provides properties as syntactic sugar for a common pattern in which a pair of methods, accessor (getter) and mutator (setter) encapsulate operations on a single attribute of a class.Full type reflection and discovery is available.C# currently (as of version 4.0) has 77 reserved words.

Checked exceptions are not present in C# (in contrast to java). This has been a conscious decision based on the issues of scalability and versionability.

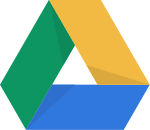
**GOOGLE SUITE**

**G Suite** is a suite of [cloud computing](https://en.wikipedia.org/wiki/Cloud_computing), [productivity](https://en.wikipedia.org/wiki/Productivity_software) and [collaboration](https://en.wikipedia.org/wiki/Collaborative_software) tools, [software](https://en.wikipedia.org/wiki/Software) and products developed by [Google Cloud](https://en.wikipedia.org/wiki/Google_Cloud_Platform), first launched on August 28, 2006 as **Google Apps for Your Domain**. G Suite comprises [Gmail](https://en.wikipedia.org/wiki/Gmail), [Hangouts](https://en.wikipedia.org/wiki/Google_Hangouts), [Calendar](https://en.wikipedia.org/wiki/Google_Calendar), and [Currents](https://en.wikipedia.org/wiki/Google_Currents) for communication; [Drive](https://en.wikipedia.org/wiki/Google_Drive) for storage; [Docs](https://en.wikipedia.org/wiki/Google_Docs), [Sheets](https://en.wikipedia.org/wiki/Google_Sheets), [Slides](https://en.wikipedia.org/wiki/Google_Slides), [Keep](https://en.wikipedia.org/wiki/Google_Keep), [Forms](https://en.wikipedia.org/wiki/Google_Forms), and [Sites](https://en.wikipedia.org/wiki/Google_Sites) for productivity and collaboration; and, depending on the plan, an Admin panel and Vault for managing users and the services. It also includes the digital [interactive whiteboard](https://en.wikipedia.org/wiki/Interactive_whiteboard) [Jamboard](https://en.wikipedia.org/wiki/Jamboard) and the app development platform [App Maker](https://en.wikipedia.org/wiki/Google_App_Maker).

While these services are free to use for consumers, G Suite adds enterprise features such as custom email addresses at a [domain](https://en.wikipedia.org/wiki/Domain_name) (@yourcompany.com), option for unlimited cloud storage (depending on plan and number of members), additional administrative tools and advanced settings, as well as 24/7 phone and email support.

Being based in Google's data centers, data and information is saved instantly and then synchronized to other data centers for backup purposes. Unlike the free, consumer-facing services, G Suite users do not see advertisements while using the services, and information and data in G Suite accounts do not get used for advertisement purposes. Furthermore, G Suite administrators can fine-tune security and privacy settings.

**GOOGLE DRIVE**



**Google Drive** is a [file storage](https://en.wikipedia.org/wiki/File_hosting_service) and [synchronization service](https://en.wikipedia.org/wiki/File_synchronization) developed by [Google](https://en.wikipedia.org/wiki/Google). Launched on April 24, 2012, Google Drive allows users to store files on their servers, synchronize files across devices, and [share files](https://en.wikipedia.org/wiki/File_sharing). In addition to a [website](https://en.wikipedia.org/wiki/Web_application), Google Drive offers apps with offline capabilities for [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) and [macOS](https://en.wikipedia.org/wiki/MacOS) computers, and [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) and [iOS](https://en.wikipedia.org/wiki/IOS) smartphones and tablets. Google Drive encompasses [Google Docs](https://en.wikipedia.org/wiki/Google_Docs), [Google Sheets](https://en.wikipedia.org/wiki/Google_Sheets), and [Google Slides](https://en.wikipedia.org/wiki/Google_Slides), which are a part of an [office suite](https://en.wikipedia.org/wiki/Office_suite) that permits collaborative editing of documents, spreadsheets, presentations, drawings, forms, and more. Files created and edited through the office suite are saved in Google Drive.

Google Drive offers users 15 [gigabytes](https://en.wikipedia.org/wiki/Gigabyte) of free storage through [Google One](https://en.wikipedia.org/wiki/Google_One). Google One also offers 100 gigabytes, 200 gigabytes, 2 [terabytes](https://en.wikipedia.org/wiki/Terabyte), 10 terabytes, 20 terabytes, and 30 terabytes offered through optional paid plans. Files uploaded can be up to 5 terabytes in size. Users can change privacy settings for individual files and folders, including enabling sharing with other users or making content public. On the website, users can search for an image by describing its visuals, and use [natural language](https://en.wikipedia.org/wiki/Natural_language_processing) to find specific files, such as "find my budget spreadsheet from last December".

The website and Android app offer a Backups section to see what Android devices have data backed up to the service, and a completely overhauled computer app released in July 2017 allows for backing up specific folders on the user's computer. A Quick Access feature can [intelligently predict](https://en.wikipedia.org/wiki/Machine_learning) the files users need.

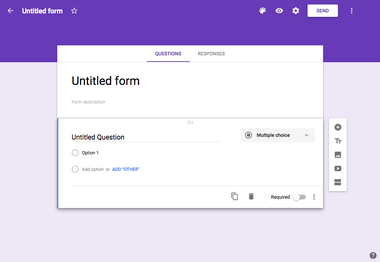
Google Drive is a key component of [G Suite](https://en.wikipedia.org/wiki/G_Suite), Google's monthly subscription offering for businesses and organizations. As part of select G Suite plans, Drive offers unlimited storage, advanced file audit reporting, enhanced administration controls, and greater collaboration tools for teams.

Following the launch of the service, Google Drive's privacy policy was heavily criticized by some members of the media. Google has one set of Terms of Service and Privacy Policy agreements that cover all of its services, meaning that the language in the agreements grants the company broad rights to reproduce, use, and create [derivative works](https://en.wikipedia.org/wiki/Derivative_work) from content stored on Google Drive. While the policies also confirm that users retain intellectual property rights, privacy advocates raised concerns that the licenses grant Google the right to use the information and data to customize the advertising and other services Google provides. In contrast, other members of the media noted that the agreements were no worse than those of competing cloud storage services, but that the competition uses "more artful language" in the agreements, and also stated that Google needs the rights in order to "move files around on its servers, cache your data, or make image thumbnails".

As of March 2017, Google Drive has 800 million active users, and as of September 2015, it has over one million organizational paying users. As of May 2017, there are over two trillion files stored on the service.

**GOOGLE FORMS**





**Google Forms** is a survey administration app that is included in the [Google Drive](https://en.wikipedia.org/wiki/Google_Drive) [office suite](https://en.wikipedia.org/wiki/Office_suite) along with [Google Docs](https://en.wikipedia.org/wiki/Google_Docs), [Google Sheets](https://en.wikipedia.org/wiki/Google_Sheets), and [Google Slides](https://en.wikipedia.org/wiki/Google_Slides). Forms features all of the collaboration and sharing features found in Docs, Sheets, and Slides.

**GOOGLE SHEETS**

****

**Google Sheets** is a [spreadsheet](https://en.wikipedia.org/wiki/Spreadsheet) program included as part of a free, [web-based](https://en.wikipedia.org/wiki/Web_application) software [office suite](https://en.wikipedia.org/wiki/Office_suite) offered by [Google](https://en.wikipedia.org/wiki/Google) within its [Google Drive](https://en.wikipedia.org/wiki/Google_Drive) service. The service also includes [Google Docs](https://en.wikipedia.org/wiki/Google_Docs) and [Google Slides](https://en.wikipedia.org/wiki/Google_Slides), a [word processor](https://en.wikipedia.org/wiki/Word_processor) and [presentation program](https://en.wikipedia.org/wiki/Presentation_program) respectively. Google Sheets is available as a [web application](https://en.wikipedia.org/wiki/Web_application), [mobile app](https://en.wikipedia.org/wiki/Mobile_app) for [Android](https://en.wikipedia.org/wiki/Android_(operating_system)), [iOS](https://en.wikipedia.org/wiki/IOS), [Windows](https://en.wikipedia.org/wiki/Windows), [BlackBerry](https://en.wikipedia.org/wiki/BlackBerry), and as a [desktop application](https://en.wikipedia.org/wiki/Desktop_application) on Google's [ChromeOS](https://en.wikipedia.org/wiki/Chrome_OS). The app is compatible with [Microsoft Excel](https://en.wikipedia.org/wiki/Microsoft_Excel) file formats. The app allows users to create and edit files online while collaborating with other users in real-time. Edits are tracked by users with a revision history presenting changes. An editor's position is highlighted with an editor-specific color and cursor and a permissions system regulates what users can do. Updates have introduced features using [machine learning](https://en.wikipedia.org/wiki/Machine_learning), including "Explore", offering answers based on [natural language](https://en.wikipedia.org/wiki/Natural_language_processing) questions in a spreadsheet.

**APPENDIX B**

**SOURCE CODE**

**HOME.ASPX**

<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="\_Default" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

<style type="text/css">

.style1

{

width: 100%;

height: 289px;

}

</style>

</asp:Content>

<asp:Content ID="Content2" runat="server"

contentplaceholderid="ContentPlaceHolder1">

<p>

<br />

</p>

<p>

</p>

<table cellspacing="0" class="style1">

<tr>

<td>

<div id="slider" class="style6">

<div id="mask" class="style5">

<ul>

<li id="first" class="firstanimation">

<a href="#">

<img src="NewFolder1/HAWEI.jfif" alt="Cashback MRG Forex 2016" />

</a>

</li>

<li id="second" class="secondanimation">

<a href="#">

<img src="NewFolder1/REDMI.jfif" alt="TF Trading Contest" />

</a>

</li>

<li id="third" class="thirdanimation">

<a href="#">

<img src="NewFolder1/S11.jfif" alt="TF Protection" class="style4" />

</a>

</li>

<li id="fourth" class="fourthanimation">

<a href="#">

<img src="NewFolder1/F11.jfif" alt="TF Trading Contest" />

</a>

</li>

<li id="fifth" class="fifthanimation">

<a href="#">

<img src="NewFolder1/S1.jfif" alt="TF Protection" class="style3" />

</a>

</li>

</ul>

</div>

<div class="progress-bar"></div>

</div>

<style type ="text/css" >

#slider

{

background-color :White;

border: 5px solid #eaeaea;

box-shadow: 1px 1px 5px rgba(0, 0, 0, 0.7);

height: 200px;

width: 1024px;

overflow: visible;

position: relative;

}

#mask {

overflow: hidden;

height: 200px;

}

#mask img {

width: 100%;

height: 100%;

}

#slider ul {

margin: 0;

padding: 0;

position: relative;

}

#slider li {

width: 1024px;

height: 200px;

position: absolute;

top: -325px;

list-style: none;

}

#slider li.firstanimation {

-moz-animation: cycle 30s linear infinite;

-webkit-animation: cycle 30s linear infinite;

}

#slider li.secondanimation {

-moz-animation: cycletwo 30s linear infinite;

-webkit-animation: cycletwo 30s linear infinite;

}

#slider li.thirdanimation {

-moz-animation: cyclethree 30s linear infinite;

-webkit-animation: cyclethree 30s linear infinite;

}

#slider li.fourthanimation {

-moz-animation:cyclefour 30s linear infinite;

-webkit-animation:cyclefour 30s linear infinite;

}

#slider li.fifthanimation {

-moz-animation:cyclefive 30s linear infinite;

-webkit-animation:cyclefive 30s linear infinite;

}

#slider:hover li,

#slider:hover .progress-bar {

-moz-animation-play-state: paused;

-webkit-animation-play-state: paused;

}

/\* PROGRESS BAR \*/

.progress-bar {

position: relative;

top: -5px;

width: 1024px;

height: 5px;

background: green;

-moz-animation: fullexpand 30s ease-out infinite;

-webkit-animation: fullexpand 30s ease-out infinite;

}

/\* ANIMATION \*/

@-moz-keyframes cycle {

0% { top:0px; }

4% { top:0px; }

16% { top:0px; opacity:1; z-index:0; }

20% { top:325px; opacity:0; z-index:0; }

21% { top:-325px; opacity:0; z-index:-1; }

92% { top:-325px; opacity:0; z-index:0; }

96% { top:-325px; opacity:0; }

100%{ top:0px; opacity:1; }

}

@-moz-keyframes cycletwo {

0% { top:-325px; opacity:0; }

16% { top:-325px; opacity:0; }

20% { top:0px; opacity:1; }

24% { top:0px; opacity:1; }

36% { top:0px; opacity:1; z-index:0; }

40% { top:325px; opacity:0; z-index:0; }

41% { top:-325px; opacity:0; z-index:-1; }

100%{ top:-325px; opacity:0; z-index:-1; }

}

@-moz-keyframes cyclethree {

0% { top:-325px; opacity:0; }

36% { top:-325px; opacity:0; }

40% { top:0px; opacity:1; }

44% { top:0px; opacity:1; }

56% { top:0px; opacity:1; }

60% { top:325px; opacity:0; z-index:0; }

61% { top:-325px; opacity:0; z-index:-1; }

100%{ top:-325px; opacity:0; z-index:-1; }

}

@-moz-keyframes cyclefour {

0% { top:-325px; opacity:0; }

56% { top:-325px; opacity:0; }

60% { top:0px; opacity:1; }

64% { top:0px; opacity:1; }

76% { top:0px; opacity:1; z-index:0; }

80% { top:325px; opacity:0; z-index:0; }

81% { top:-325px; opacity:0; z-index:-1; }

100%{ top:-325px; opacity:0; z-index:-1; }

}

@-moz-keyframes cyclefive {

0% { top:-325px; opacity:0; }

76% { top:-325px; opacity:0; }

80% { top:0px; opacity:1; }

84% { top:0px; opacity:1; }

96% { top:0px; opacity:1; z-index:0; }

100%{ top:325px; opacity:0; z-index:0; }

}

@-webkit-keyframes cycle {

0% { top:0px; }

4% { top:0px; }

16% { top:0px; opacity:1; z-index:0; }

20% { top:325px; opacity:0; z-index:0; }

21% { top:-325px; opacity:0; z-index:-1; }

50% { top:-325px; opacity:0; z-index:-1; }

92% { top:-325px; opacity:0; z-index:0; }

96% { top:-325px; opacity:0; }

100%{ top:0px; opacity:1; }

}

@-webkit-keyframes cycletwo {

0% { top:-325px; opacity:0; }

16% { top:-325px; opacity:0; }

20% { top:0px; opacity:1; }

24% { top:0px; opacity:1; }

36% { top:0px; opacity:1; z-index:0; }

40% { top:325px; opacity:0; z-index:0; }

41% { top:-325px; opacity:0; z-index:-1; }

100%{ top:-325px; opacity:0; z-index:-1; }

}

@-webkit-keyframes cyclethree {

0% { top:-325px; opacity:0; }

36% { top:-325px; opacity:0; }

40% { top:0px; opacity:1; }

44% { top:0px; opacity:1; }

56% { top:0px; opacity:1; z-index:0; }

60% { top:325px; opacity:0; z-index:0; }

61% { top:-325px; opacity:0; z-index:-1; }

100%{ top:-325px; opacity:0; z-index:-1; }

}

@-webkit-keyframes cyclefour {

0% { top:-325px; opacity:0; }

56% { top:-325px; opacity:0; }

60% { top:0px; opacity:1; }

64% { top:0px; opacity:1; }

76% { top:0px; opacity:1; z-index:0; }

80% { top:325px; opacity:0; z-index:0; }

81% { top:-325px; opacity:0; z-index:-1; }

100%{ top:-325px; opacity:0; z-index:-1; }

}

@-webkit-keyframes cyclefive {

0% { top:-325px; opacity:0; }

76% { top:-325px; opacity:0; }

80% { top:0px; opacity:1; }

84% { top:0px; opacity:1; }

96% { top:0px; opacity:1; z-index:0; }

100%{ top:325px; opacity:0; z-index:0; }

}

/\* ANIMATION BAR \*/

@-moz-keyframes fullexpand {

0%, 20%, 40%, 60%, 80%, 100% { width:0%; opacity:0; }

4%, 24%, 44%, 64%, 84% { width:0%; opacity:0.3; }

16%, 36%, 56%, 76%, 96% { width:100%; opacity:0.7; }

17%, 37%, 57%, 77%, 97% { width:100%; opacity:0.3; }

18%, 38%, 58%, 78%, 98% { width:100%; opacity:0; }

}

@-webkit-keyframes fullexpand {

0%, 20%, 40%, 60%, 80%, 100% { width:0%; opacity:0; }

4%, 24%, 44%, 64%, 84% { width:0%; opacity:0.3; }

16%, 36%, 56%, 76%, 96% { width:100%; opacity:0.7; }

17%, 37%, 57%, 77%, 97% { width:100%; opacity:0.3; }

18%, 38%, 58%, 78%, 98% { width:100%; opacity:0; }

}

.style5

{

width: 1017px;

height: 173px;

margin-left: 0px;

}

.style6

{

left: 220px;

top: -21px;

height: 245px;

width: 1018px;

}

.style7

{

width: 100%;

}

</style>

&nbsp;</td>

</tr>

</table>

<p>

</p>

<p>

</p>

<p>

</p>

</asp:Content>

**MOBILE.ASPX**

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default2.aspx.cs" Inherits="Default2" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

<body>

<form id="form1" runat="server">

<link href='https://fonts.googleapis.com/css?family=Lato:300,400' rel='stylesheet' type='text/css'>

<link rel="stylesheet" href="https://weloveiconfonts.com/api/?family=entypo">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/normalize/5.0.0/normalize.min.css">

<link rel='stylesheet' href='https://s3-us-west-2.amazonaws.com/s.cdpn.io/5175/utf-latest.min.css'>

<script src="https://code.jquery.com/jquery-3.1.1.slim.min.js" integrity="sha384-A7FZj7v+d/sdmMqp/nOQwliLvUsJfDHW+k9Omg/a/EheAdgtzNs3hpfag6Ed950n" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/tether/1.4.0/js/tether.min.js" integrity="sha384-DztdAPBWPRXSA/3eYEEUWrWCy7G5KFbe8fFjk5JAIxUYHKkDx6Qin1DkWx51bBrb" crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/js/bootstrap.min.js" integrity="sha384-vBWWzlZJ8ea9aCX4pEW3rVHjgjt7zpkNpZk+02D9phzyeVkE+jo0ieGizqPLForn" crossorigin="anonymous"></script>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/css/bootstrap.min.css" integrity="sha384-rwoIResjU2yc3z8GV/NPeZWAv56rSmLldC3R/AZzGRnGxQQKnKkoFVhFQhNUwEyJ" crossorigin="anonymous">

<link rel="apple-touch-icon" type="image/png" href="https://static.codepen.io/assets/favicon/apple-touch-icon-5ae1a0698dcc2402e9712f7d01ed509a57814f994c660df9f7a952f3060705ee.png" />

<div>

<div class="container">

<header class="header">

<h3 class="header-logo">MOBILES</h3>

<nav class="header-nav">

<ul>

<li><a href="Default.aspx" >Home</a></li>

<li><a href="Default5.aspx" >OPPO</a></li>

<li><a href="Default6.aspx" >REDMI</a></li>

<li><a href="Default7.aspx" >HAWEI</a></li>

</ul>

</nav>

</header>

<STYLE type ="text/css" >

\* {

box-sizing: border-box;

}

body {

margin: 0;

padding: 0;

font-family: 'Open Sans', sans-serif;

font-size: 16px;

background: cornsilk;

color: #333;

}

a {

text-decoration: none;

}

.button {

padding: 10px;

border-radius: 5px;

display: inline-block;

background: steelblue;

color: white;

font-size: 12px;

font-weight: bold;

}

.container {

max-width: 960px;

width: 100%;

margin: 0 auto;

}

.header {

overflow: hidden;

height: 65px;

}

.header a {

color: inherit;

}

.header a:hover {

font-weight: bold;

}

.header-logo {

float: left;

}

.header-nav {

float: right;

}

.header-nav ul li {

display: inline-block;

margin: 0 20px;

}

</STYLE>

<br />

<br />

<br />

<br />

<br />

<!-- Nav -->

<nav class="navbar navbar-inverse bg-inverse fixed-top bg-faded">

<div class="row">

<div class="col">

<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#cart">Cart (<span class="total-count"></span>)</button><button class="clear-cart btn btn-danger">Clear Cart</button></div>

</div>

</nav>

<!-- Main -->

<div class="container">

<div class="row">

<div class="col">

<div class="card" style="width: 20rem;">

<br />

<img class="style1" src="NewFolder3/S1.jfif" alt="Card image cap">

<div class="card-block">

<h4 class="card-title">S1</h4>

<p class="card-text">Price: $98</p>

<a href="#" data-name="S1" data-price="15000" class="add-to-cart btn btn-primary">Add to cart</a>

<a href="">COD</a>

<a href="#" >DETAILS</a>

</div>

</div>

</div>

<div class="col">

<div class="card" style="width: 20rem;">

<img class="style2" src="NewFolder3/V15.jfif" alt="Card image cap"><br />

<div class="card-block">

<h4 class="card-title">V15</h4>

<p class="card-text">Price: $98</p>

<a href="#" data-name="V15" data-price="15000" class="add-to-cart btn btn-primary">Add to cart</a>

<a href="#">COD</a>

<a href="#" >DETAILS</a>

</div>

</div>

</div>

<div class="col">

<div class="card" style="width: 20rem;">

<img class="style3" src="NewFolder3/V17.jfif" alt="Card image cap">

<div class="card-block">

<h4 class="card-title">V17</h4>

<p class="card-text">Price: $100</p>

<a href="#" data-name="V17" data-price="20000" class="add-to-cart btn btn-primary">Add to cart</a>

<a href="#" >COD</a>

<a href="#" >DETAILS</a>

</div>

</div>

</div>

<div class="col">

<div class="card" style="width: 20rem;">

<img class="style4" src="NewFolder3/Z1.jfif" alt="Card image cap">

<div class="card-block">

<h4 class="card-title">Z1 PRO</h4>

<p class="card-text">Price: $100</p

<a href="#" data-name="Z1 PRO" data-price="20000" class="add-to-cart btn btn-primary">Add to cart</a>

<a href="#" >COD</a>

<a href="#" >DETAILS</a>

</div>

</div>

</div>

</div>

</div>

<!-- Modal -->

<div class="modal fade" id="cart" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">

<div class="modal-dialog modal-lg" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="exampleModalLabel">Cart</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

<table class="show-cart table">

</table>

<div>Total price: <span class="total-cart"></span></div>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>

<button type="button" class="btn btn-primary">Order now</button>

</div>

</div>

</div>

</div>

<style type ="text/css" >

body {

padding-top: 80px;

}

.show-cart li {

display: flex;

}

.card {

margin-bottom: 20px;

}

.card-img-top {

width: 200px;

height: 200px;

align-self: center;

}

.style1

{

width: 120px;

height: 200px;

align-self: center;

}

.style2

{

width: 132px;

height: 200px;

align-self: center;

}

.style3

{

width: 123px;

height: 200px;

align-self: center;

}

.style4

{

width: 107px;

height: 200px;

align-self: center;

}

</style>

<script type ="text/javascript" >

var shoppingCart = (function () {

// =============================

// Private methods and propeties

// =============================

cart = [];

// Constructor

function Item(name, price, count) {

this.name = name;

this.price = price;

this.count = count;

}

// Save cart

function saveCart() {

sessionStorage.setItem('shoppingCart', JSON.stringify(cart));

}

// Load cart

function loadCart() {

cart = JSON.parse(sessionStorage.getItem('shoppingCart'));

}

if (sessionStorage.getItem("shoppingCart") != null) {

loadCart();

}

// =============================

// Public methods and propeties

// =============================

var obj = {};

// Add to cart

obj.addItemToCart = function (name, price, count) {

for (var item in cart) {

if (cart[item].name === name) {

cart[item].count++;

saveCart();

return;

}

}

var item = new Item(name, price, count);

cart.push(item);

saveCart();

}

// Set count from item

obj.setCountForItem = function (name, count) {

for (var i in cart) {

if (cart[i].name === name) {

cart[i].count = count;

break;

}

}

};

// Remove item from cart

obj.removeItemFromCart = function (name) {

for (var item in cart) {

if (cart[item].name === name) {

cart[item].count--;

if (cart[item].count === 0) {

cart.splice(item, 1);

}

break;

}

}

saveCart();

}

// Remove all items from cart

obj.removeItemFromCartAll = function (name) {

for (var item in cart) {

if (cart[item].name === name) {

cart.splice(item, 1);

break;

}

}

saveCart();

}

// Clear cart

obj.clearCart = function () {

cart = [];

saveCart();

}

// Count cart

obj.totalCount = function () {

var totalCount = 0;

for (var item in cart) {

totalCount += cart[item].count;

}

return totalCount;

}

// Total cart

obj.totalCart = function () {

var totalCart = 0;

for (var item in cart) {

totalCart += cart[item].price \* cart[item].count;

}

return Number(totalCart.toFixed(2));

}

// List cart

obj.listCart = function () {

var cartCopy = [];

for (i in cart) {

item = cart[i];

itemCopy = {};

for (p in item) {

itemCopy[p] = item[p];

}

itemCopy.total = Number(item.price \* item.count).toFixed(2);

cartCopy.push(itemCopy)

}

return cartCopy;

}

// cart : Array

// Item : Object/Class

// addItemToCart : Function

// removeItemFromCart : Function

// removeItemFromCartAll : Function

// clearCart : Function

// countCart : Function

// totalCart : Function

// listCart : Function

// saveCart : Function

// loadCart : Function

return obj;

})();

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Triggers / Events

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Add item

$('.add-to-cart').click(function (event) {

event.preventDefault();

var name = $(this).data('name');

var price = Number($(this).data('price'));

shoppingCart.addItemToCart(name, price, 1);

displayCart();

});

// Clear items

$('.clear-cart').click(function () {

shoppingCart.clearCart();

displayCart();

});

function displayCart() {

var cartArray = shoppingCart.listCart();

var output = "";

for (var i in cartArray) {

output += "<tr>"

+ "<td>" + cartArray[i].name + "</td>"

+ "<td>(" + cartArray[i].price + ")</td>"

+ "<td><div class='input-group'><button class='minus-item input-group-addon btn btn-primary' data-name=" + cartArray[i].name + ">-</button>"

+ "<input type='number' class='item-count form-control' data-name='" + cartArray[i].name + "' value='" + cartArray[i].count + "'>"

+ "<button class='plus-item btn btn-primary input-group-addon' data-name=" + cartArray[i].name + ">+</button></div></td>"

+ "<td><button class='delete-item btn btn-danger' data-name=" + cartArray[i].name + ">X</button></td>"

+ " = "

+ "<td>" + cartArray[i].total + "</td>"

+ "</tr>";

}

$('.show-cart').html(output);

$('.total-cart').html(shoppingCart.totalCart());

$('.total-count').html(shoppingCart.totalCount());

}

// Delete item button

$('.show-cart').on("click", ".delete-item", function (event) {

var name = $(this).data('name')

shoppingCart.removeItemFromCartAll(name);

displayCart();

})

// -1

$('.show-cart').on("click", ".minus-item", function (event) {

var name = $(this).data('name')

shoppingCart.removeItemFromCart(name);

displayCart();

})

// +1

$('.show-cart').on("click", ".plus-item", function (event) {

var name = $(this).data('name')

shoppingCart.addItemToCart(name);

displayCart();

})

// Item count input

$('.show-cart').on("change", ".item-count", function (event) {

var name = $(this).data('name');

var count = Number($(this).val());

shoppingCart.setCountForItem(name, count);

displayCart();

});

displayCart();

</script><br />

<br />

</div>

</form>

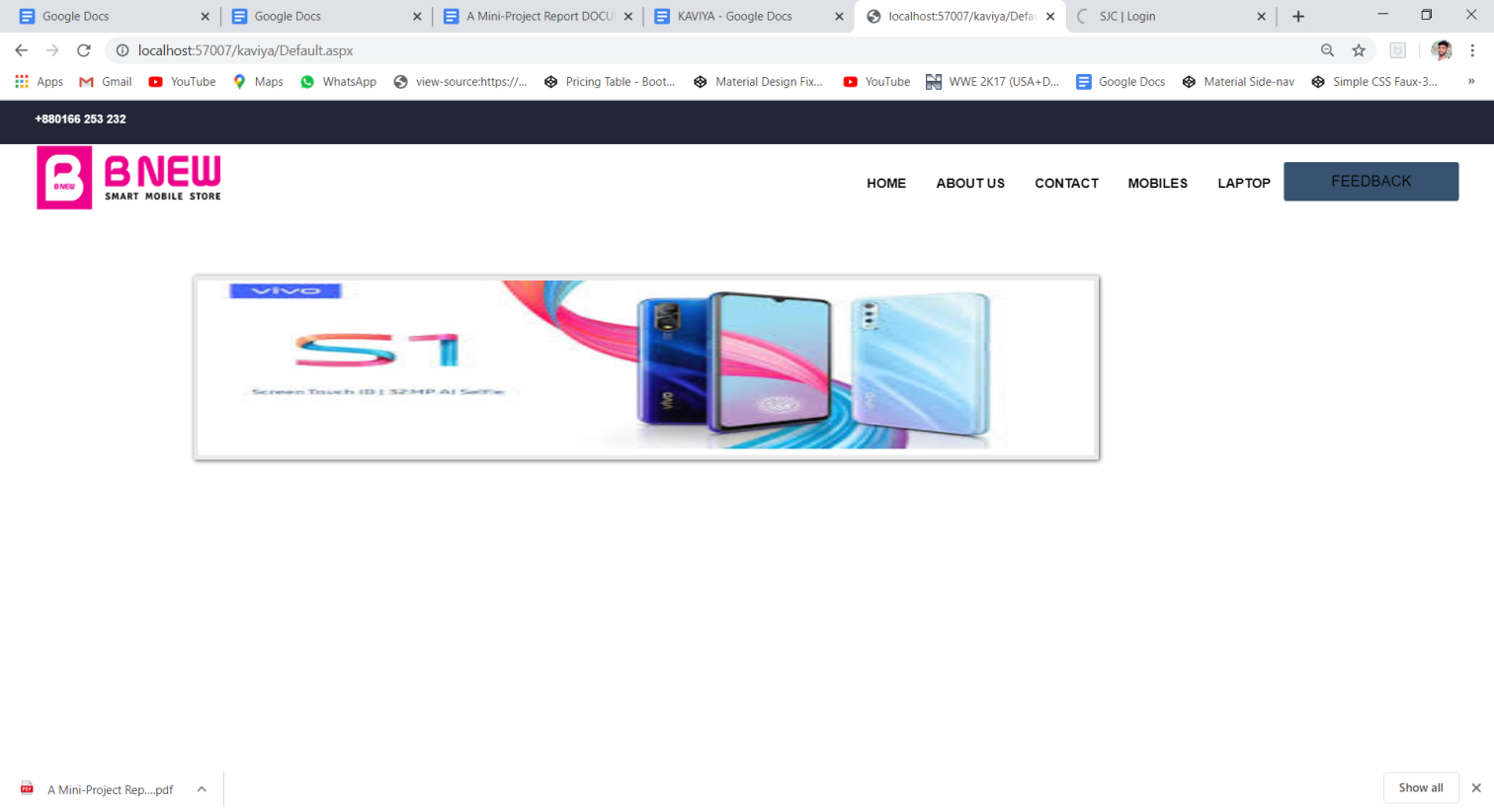
</body>

</html>

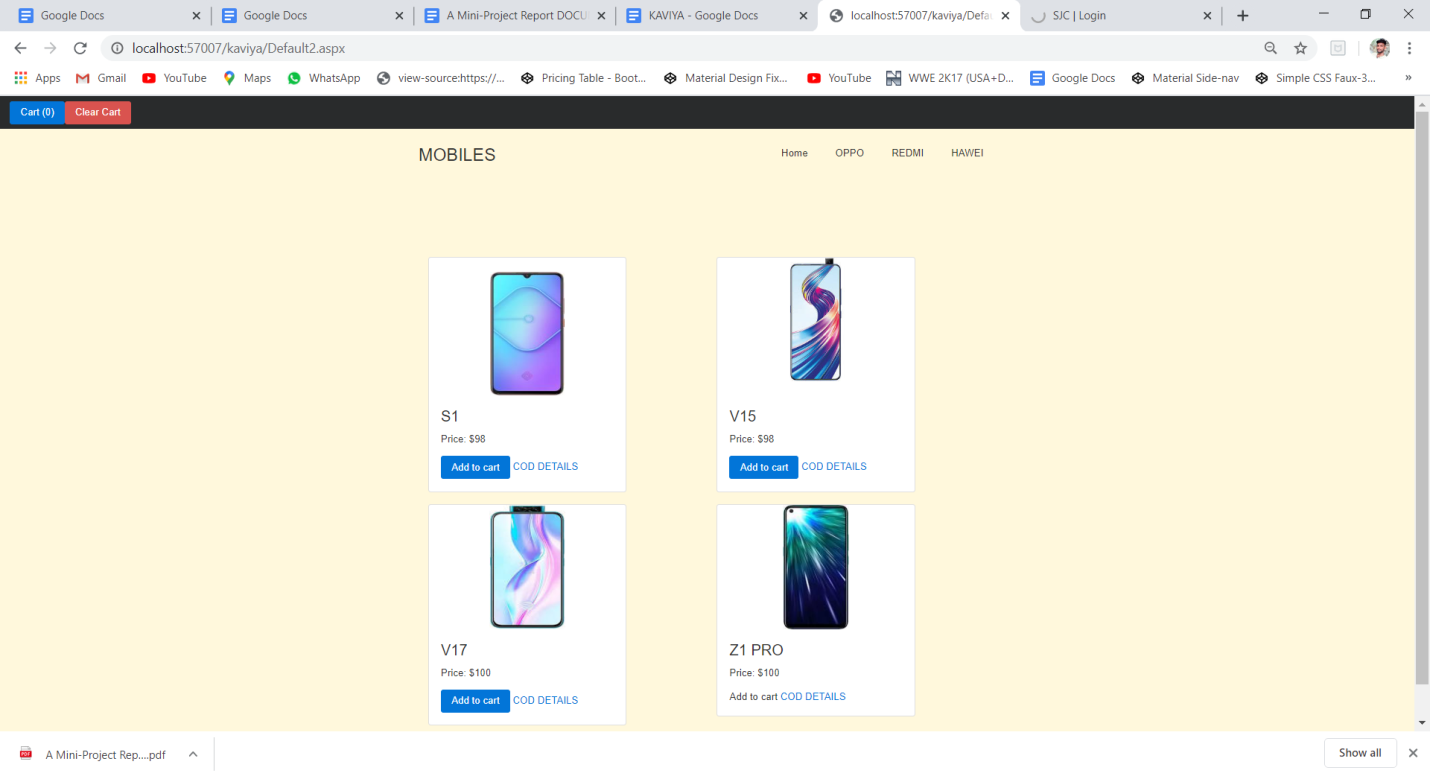
**APPENDIX C**

**SCREENSHOTS**

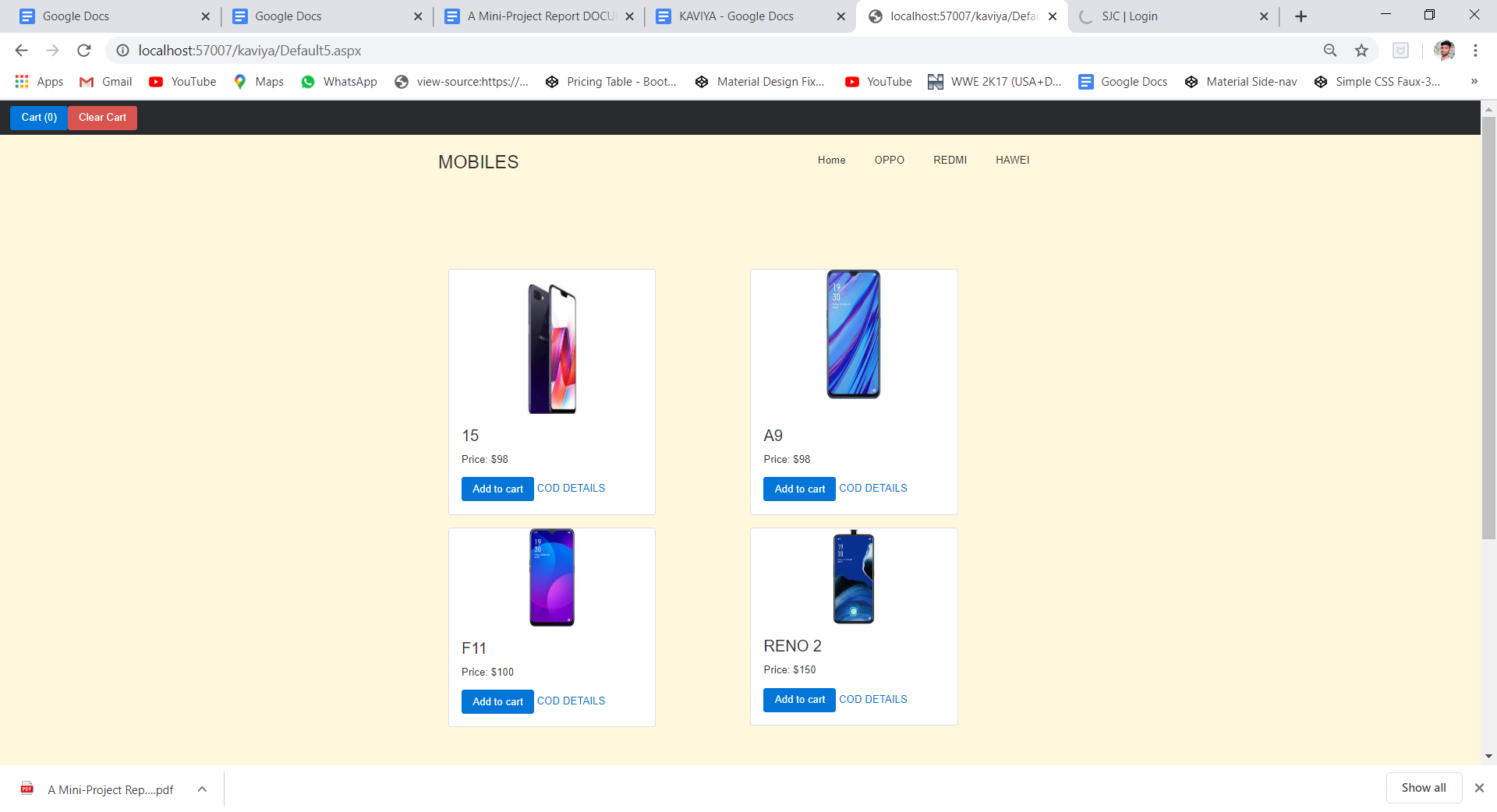
**HOME PAGE**

****

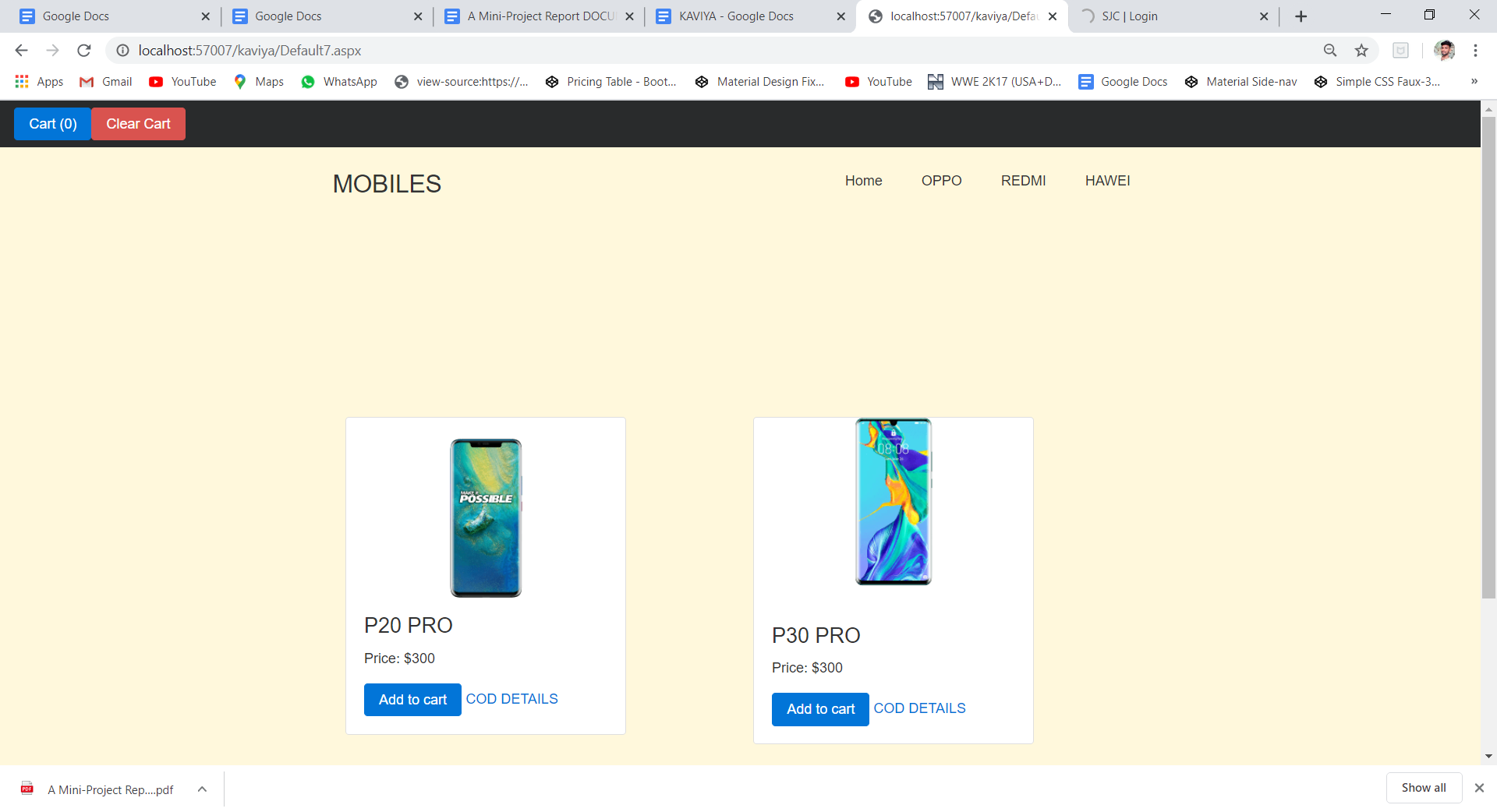
**VIVO MOBILES**

****

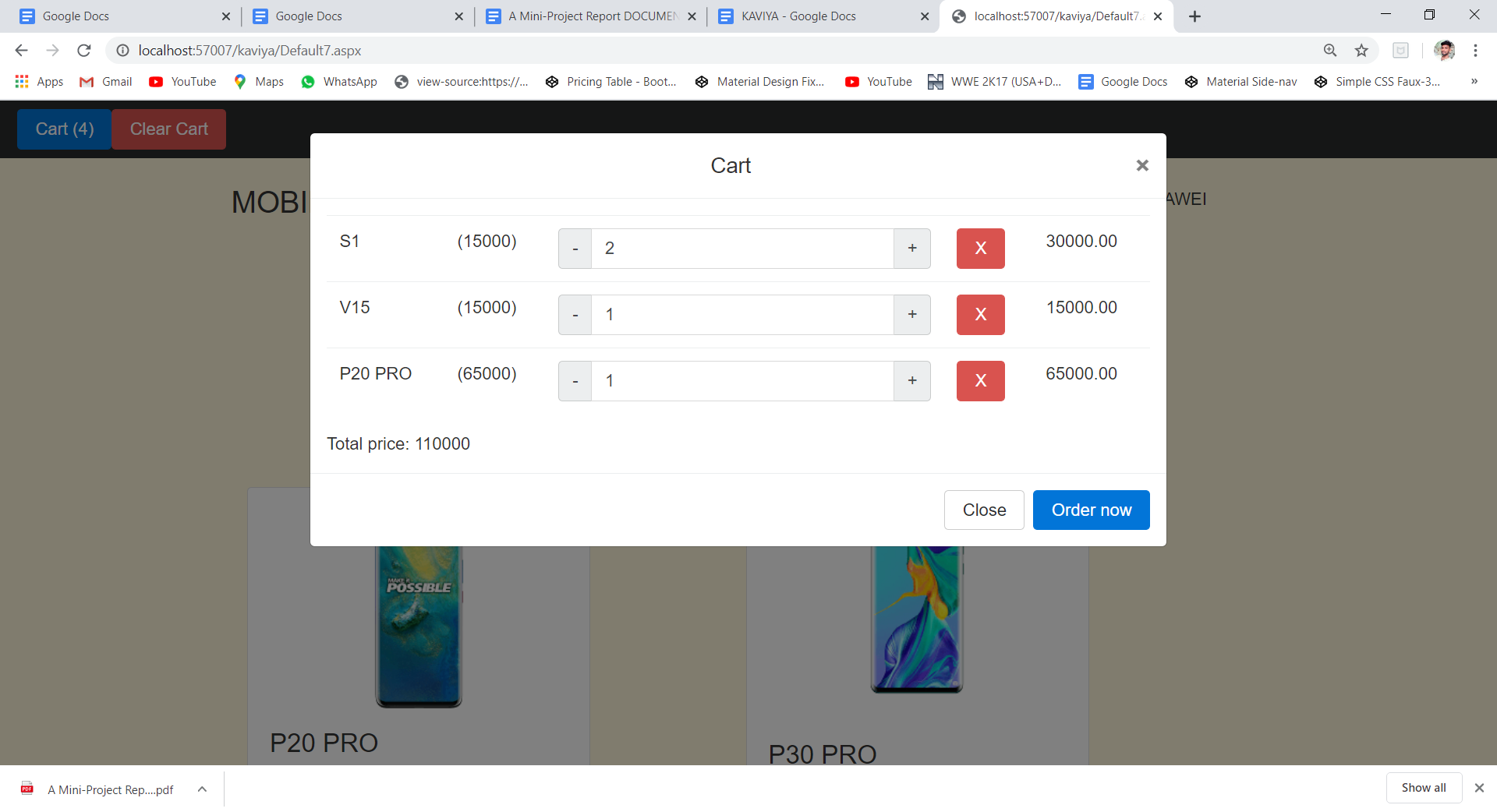
**OPPO MOBILES**

****

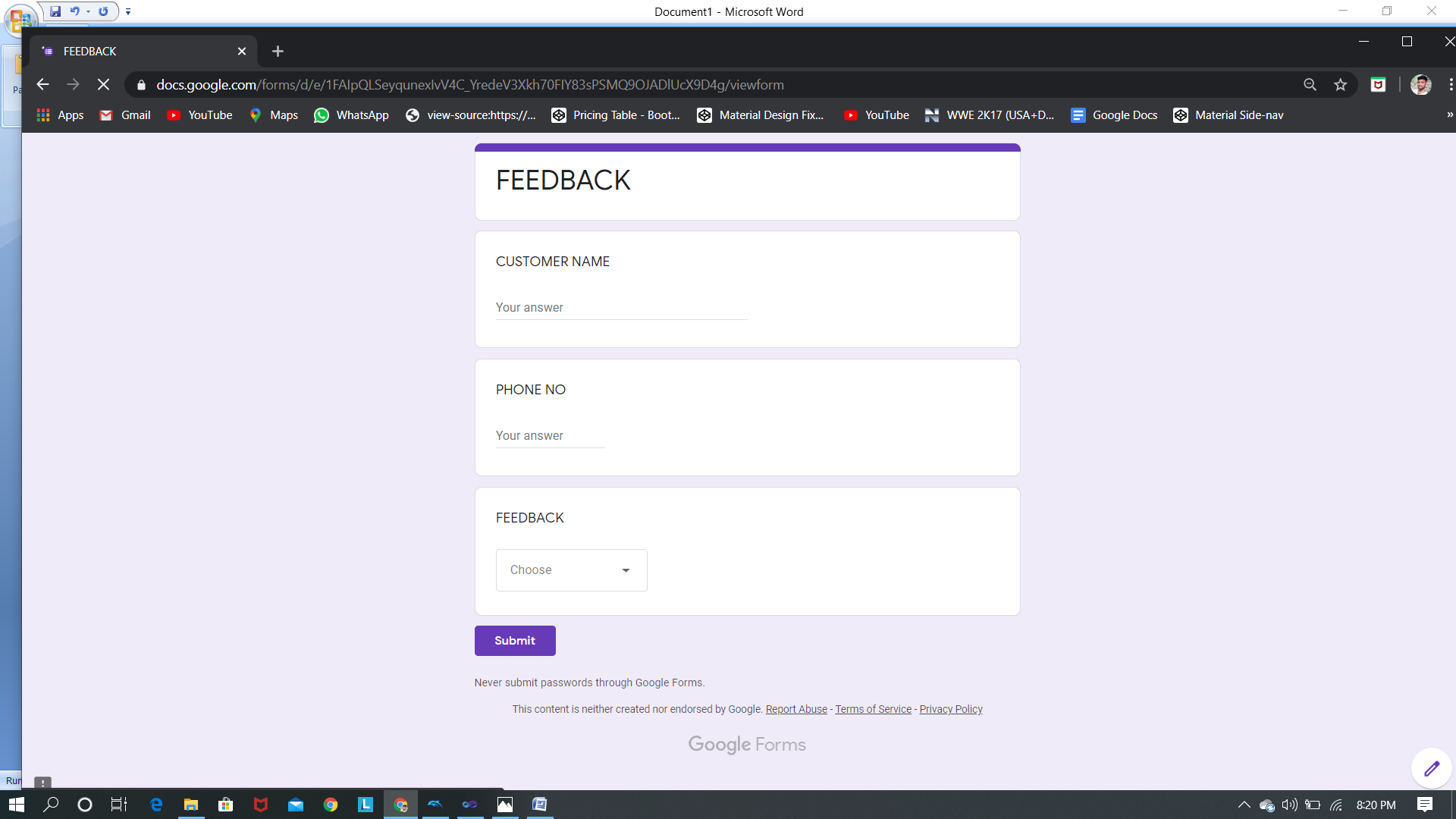
**HAWEI MOBILES**

****

**ADD TO CART**

****

**FEEDBACK**

****