

**Sankalpa Neupane**

**DEVELOPING A STATIC WEBSITE AND DEPLOYING IT TO  
HEROKU**

**Thesis  
CENTRIA UNIVERSITY OF APPLIED SCIENCES  
Information Technology  
November 2020**

**ABSTRACT**

<b>Centria University of Applied Sciences</b>	<b>Date</b> November 2020	<b>Author</b> Sankalpa Neupane
<b>Degree programme</b> Information Technology		
<b>Name of thesis</b> DEVELOPING A STATIC WEBSITE AND DEPLOYING IT TO HEROKU		
<b>Instructor</b>	<b>Pages</b> 33 + 2	
<b>Supervisor</b> Jari Isohanni		
<p>A website is a most useful tool for all kind of business around the globe and the easiest way to reach out to the people. So, a good user satisfying website is a must in today's business. A website should be deployed to make it available for all the users around the world. So, a website deployment can be done using a cloud platform which allows developers to maintain, scale and test the website. This thesis is implemented in such a way that it demonstrates how a website is developed and can be deployed in the cloud platform.</p> <p>The thesis aims to present the process of building and deploying a static website on a cloud platform. The website was made up of a single page. The website was focused on front-end development. The process included designing and developing by using HTML5, CSS 3, JavaScript, and Bootstrap 4.5. The process also included the deployment of a static website to Heroku.</p> <p>Heroku is a cloud platform which allows users to build, test, maintain and scale the web applications. It also allows the user to deploy the web application from a local environment to the server. It supports many programming languages like PHP, Java, Node.js, Python, and Go.</p>		

<p><b>Key words</b> API, CLI, Cloud Platform, Deployment, Heroku, Heroku Web Applications, PaaS, Wireframing</p>
--

## **CONCEPT DEFINITIONS**

### **List of Abbreviations**

API	Application Programming Interface
AWS	Amazon Web Services
CLI	Command Line Interface
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IaaS	Infrastructure-as-a-Service
PaaS	Platform-as-a-Service
SaaS	Software-as-a-Service

**ABSTRACT**  
**CONCEPT DEFINITIONS**  
**CONTENTS**

<b>1 INTRODUCTION</b> .....	<b>1</b>
<b>2 THEORETICAL BACKGROUND</b> .....	<b>2</b>
<b>2.1 Cloud Services, App and Platform</b> .....	<b>3</b>
<b>2.2 Website as applications</b> .....	<b>4</b>
<b>2.3 Platform-as-a-Service (PaaS)</b> .....	<b>5</b>
<b>2.4 Cloud application deployment</b> .....	<b>6</b>
<b>3 HEROKU CLOUD PLATFORM</b> .....	<b>7</b>
<b>3.1 Heroku’s terminology</b> .....	<b>7</b>
<b>3.2 Platform and solution stack</b> .....	<b>10</b>
<b>3.2.1 Stack and build packs</b> .....	<b>10</b>
<b>3.2.2 Procfile, process type and dynos</b> .....	<b>10</b>
<b>3.2.3 Routing, regions and add-ons</b> .....	<b>11</b>
<b>3.3 Deployment flow</b> .....	<b>11</b>
<b>3.4 Runtime management</b> .....	<b>12</b>
<b>4 ARCHITECTURE OF A WEBSITE</b> .....	<b>13</b>
<b>4.1 HTML 5</b> .....	<b>13</b>
<b>4.2 CSS</b> .....	<b>14</b>
<b>4.3 Bootstrap</b> .....	<b>16</b>
<b>4.4 Javascript</b> .....	<b>18</b>
<b>5 IMPLEMENTATION</b> .....	<b>19</b>
<b>5.1 Website wireframing</b> .....	<b>19</b>
<b>5.2 Website structure</b> .....	<b>20</b>
<b>5.3 Development</b> .....	<b>21</b>
<b>5.4 Website deployment</b> .....	<b>27</b>
<b>5.5 Running the app on Heroku</b> .....	<b>31</b>
<b>6 CONCLUSION</b> .....	<b>33</b>
<b>REFERENCES</b> .....	<b>33</b>
<b>APPENDICES</b>	

**FIGURES**

FIGURE 1. The Cloud (Cloudflare Inc, 2020) .....	2
FIGURE 2. Heroku architecture. (Packt, 2019).....	11
FIGURE 3. HTML code (MDN, 2020) .....	13
FIGURE 4. Hello World! (Geeks for Geeks, 2019) .....	14
FIGURE 5. Inline CSS (MDN, 2020).....	15
FIGURE 6. Internal CSS (MDN, 2020).....	15
FIGURE 7. External CSS (MDN, 2020).....	15
FIGURE 8. Installing Bootstrap using npm package manager (Bootstrap, 2020).....	17
FIGURE 9. Installing Bootstrap using gem package manager (Bootstrap, 2020).....	17

FIGURE 10. Bootstrap CDN (Bootstrap, 2020) .....	17
FIGURE 11. JS compiled Bootstrap CDN (Bootstrap, 2020) .....	18
FIGURE 12. Website wireframing .....	20
FIGURE 13. Website structure .....	21
FIGURE 14. Setting up the project .....	22
FIGURE 15. HTML boilerplate .....	23
FIGURE 16. Index HTML file .....	24
FIGURE 17. HTML file without CSS .....	24
FIGURE 18. Adding CSS to HTML file .....	25
FIGURE 19. After adding CSS to HTML file .....	25
FIGURE 20. Navigation using id .....	26
FIGURE 21. JavaScript file .....	27
FIGURE 22. Git bash terminal and changing of directory .....	28
FIGURE 23. PHP inclusion using commands .....	28
FIGURE 24. Command for creating a JSON file .....	29
FIGURE 25. Command for git id and commit message .....	29
FIGURE 26. Heroku login .....	30
FIGURE 27. Login page of Heroku .....	30
FIGURE 28. Command for creating an app on Heroku .....	30
FIGURE 29. Deployment process .....	31
FIGURE 30. Heroku Dashboard .....	32
FIGURE 31. Running the web app in the servers using Heroku .....	32

## TABLES

TABLE 1. Heroku's Terminology (Ansaharju 2016) .....	9
--	---

## 1 INTRODUCTION

The rise of technology and the idea of an era of modern as well as advanced technologies has overwhelmed today's society. Firstly, due to the advancement in technology, people can go anywhere geographically. Secondly, people can get access to information and knowledge they need through the use of the Internet. The Internet is the most rapidly growing environment in case of exchanging information (Nguyen 2020). It is hard to find areas where the Internet would not have its full and comprehensive reflection.

However, it is hard to imagine the rapid growth of information exchange without web applications and webpages. Until the Mosaic browsers were developed, there were standard text and tables contained in the web pages. Later in Mosaic browsers, people were able to see the pictures in the web browsers. Till that, the web applications and websites were not so dynamic. In 1995, a new client-side scripting language called JavaScript was introduced by Netscape. After JavaScript was introduced, the history of web pages started to become dynamic. (Nguyen 2020)

Nevertheless, the concept of the web application was introduced in 1999 (Nguyen 2020). As the web-applications, websites have evolved from static web pages to dynamic websites. Nowadays, the websites are so vibrant and user-friendly that ordinary people can use them easily. People can use them to get information from around the world. The development of the web has made web applications quite popular that they are used in a variety of fields like appointment systems and instant messaging.

This thesis focuses on developing a website and how it can be deployed to Heroku. A Heroku is a cloud platform that lets companies build, deliver, monitor, and scale apps. It is such a platform that supports several programming languages. It has integrated data services and a robust ecosystem to deploy and to run modern apps. Heroku is a platform-based service on a managed container system. This thesis project was performed in the context of deploying a complete website and deploying it to Heroku. General level background information is gathered on technical details about Heroku and how to use it.

## 2 THEORETICAL BACKGROUND

Cloud implies to the servers and these servers are placed in data centres all over the world. The cloud servers are accessed over the Internet, and all the databases and software run over these servers. The cloud empowers users to get to similar documents and applications from practically any devices because computing and data storages occur in the servers in the data centres rather than locally on the user's device. This is the reason why a user can sign into their Facebook account on another device after their old device breaks and still locate their old record set up, with all their photographs, recordings, and discussion history. The process of storing data and information in a cloud is called cloud computing. Figure 1 shows how a cloud works. (Cloudflare Inc, 2020)

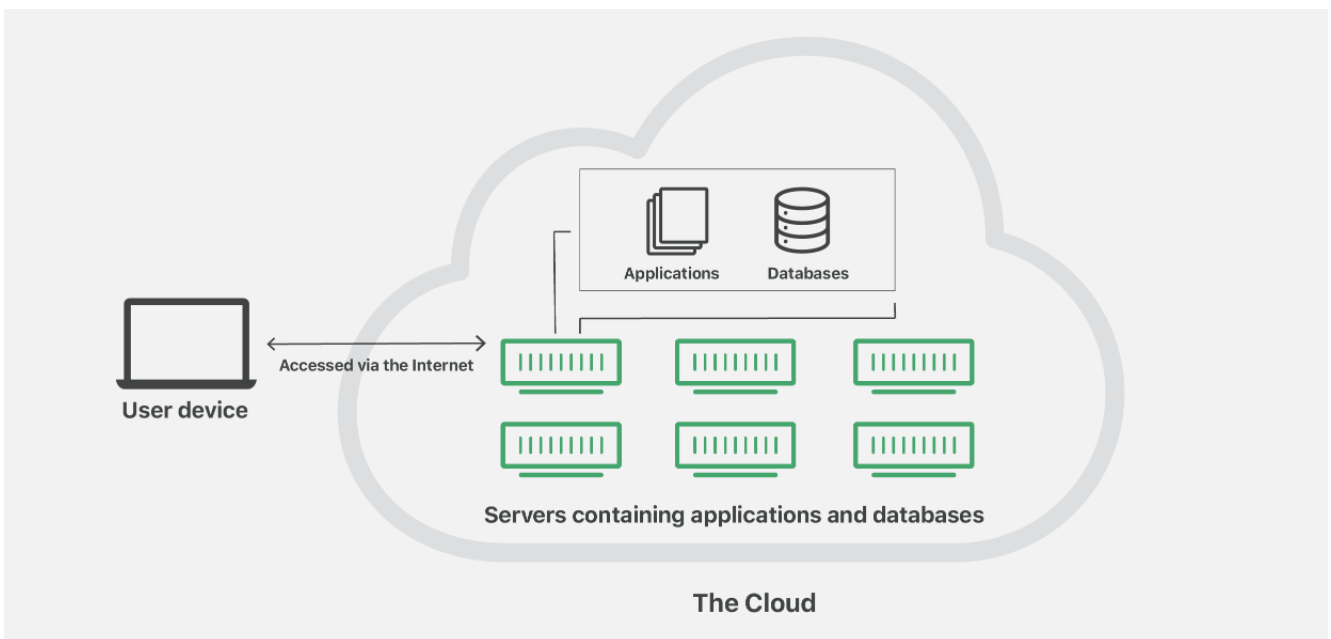


FIGURE 1. The Cloud. (Cloudflare Inc, 2020).

Cloud computing is the conveyance of various services and administration through the Internet. These assets incorporate tools and applications like data storage, networking, servers, database and software. The Cloud computing term has been used in the early 2000s, but the concept of computer as-a-service has been used previously far ago back in the 1960s. In the meantime, companies were not able to buy the personal computer to store data where they used the mainframe computers rented by the computer bureaus. As time passed PC became cheap and affordable and companies started to store their data using various media and cloud computing is one of them. The concept of renting access to computing power has resurfaced over time and the cloud computing which truly took hold of the development of software

as assistance and hyper-scale cloud was computing providers like, Amazon Web Services. (Ranger 2018)

The main concept of cloud computing is that companies do not need to buy the infrastructure to store data and information, but rather than they can rent cloud services provided by the cloud service providers. One advantage of utilizing a cloud computing service is that organizations can evade the forthright expense and multifaceted nature of possessing and keeping up their own IT framework, and rather basically pay for what they use when they use it. Cloud-based storage makes it conceivable to spare documents to a distant information base and recover them on request. (Rangers 2018)

## **2.1 Cloud Services, App and Platform**

Cloud services are software that is facilitated by third-party suppliers and made accessible to clients through the web. Cloud services encourage the progression of client information from front-end customers (for example clients' servers, tablets, work areas, workstations—any equipment on the clients' closures), through the web, to the supplier's frameworks. Clients can get to cloud administrations with just a PC, a working framework, and an organization association with the web. All software that clients access through the web without requiring extra software downloads can be viewed as a cloud service. A full cloud app is also a service of cloud services. (Red Hat, 2020)

A cloud application is an Internet-based program where a couple, or all, of the processing logic and data accumulating, is dealt with in the cloud. The clients communicate with the application using an internet browser or a mobile application, and the data processing is administered by a local device and computing solution. From the client's perspective, the cloud application continues like a standard website, where the data processing is dealt with the cloud through an API (application program interface) or a mixture of both. Cloud apps have several benefits like reduced software costs, control, security, and reduced management cost. Cloud applications have different categories, and they are Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS). (Barton 2018)

The most popular approaches to developing an app for mobile and online is by using HTML, CSS, and JavaScript. These three-programming languages help to create an interactive and browser-based web app. HTML is the structure or the foundation of the website and it adds content to the webpages. The CSS is used to give the style to a website, and it adds presentation to the content of the webpages.

JavaScript makes a website interactive and makes it functional. JavaScript add interactive elements in the content of webpages. The cloud website or app is deployed in a various cloud platform. The websites or app built on local environment can also be deployed on a cloud platform. (Red Hat, 2020)

The cloud platform are kind of PaaS. Also, if the infrastructural parts holding up the PaaS are exceptionally versatile and sharable, it may be viewed as a cloud. The best instances of PaaS clouds incorporate public cloud and managed private cloud. In the cloud, platform users can build and run apps on the cloud. The cloud service providers use their assets to create a cloud platform. Building a cloud platform requires something beyond abstracting a PC's abilities from its equipment parts, like when giving a cloud foundation. A cloud platform requires extra degrees of improvement to join innovations like containerization, organization, application programming interfaces (APIs), directing, security, and robotization. Client experience plan (UX) is likewise a significant thought to make a safe online encounter. Some cloud platforms are Heroku, Google cloud platform, Microsoft Azure, etc. (Red Hat, 2020)

## **2.2 Websites as applications**

A website is an assortment of linked web pages (in addition to their related assets) that share a unique domain name. Each page of a given site gives express connections between pages, often as an interactive bit of text, that permits the client to move to start with one page of the site then onto the next. The websites are created and maintained by different organizations, groups and individuals for different purposes. (MDN, 2020a)

A website makes it simple for clients to explore and extricate applicable data that addresses their issues. It is accessed and updated easily. A website is the best platform to advertise any kind of product or service because in today's world it is the easiest way to reach out to the people around the world (Georgiou 2019). On the other hand, a web application is a software that runs over internet browsers. In brief, a software program that can run on the internet browsers where it permits incorporation and trade of information from the customer PC to the server. Such applications utilize the assets of the servers and clients to run. They do not need a complete application to be stacked on the client PCs and the applications can run utilizing less RAM (Mont Digital 2019). The main task of the web application is to be receptive to client activities; to be intuitive and give clients the capacity to control information and make demands for various fields. As web application is also a server-side application so it uses the server-side

programming language also compared to a website where it only uses client-side programming languages. (Georgiou 2019)

Both website and web application have similarities. So, a static website can be used as a static web app. A static web application is ordinarily assembled utilizing libraries and structures like Angular, React, Svelte, Vue, or Blazor. These applications incorporate HTML, CSS, JavaScript, and picture resources that make up the application. With a traditional web server, these advantages are served from a single server close by any necessary API endpoints (Microsoft, 2020). In a static web app, the user sends a request to a web server using a browser and then the server gets the network request and sends back the information along with the web application. During this process, web application connects to a database and stores the information based on the requested character. The web application now generates an automatic HTML which is sent directly to the user's screen by the server. (Static app, 2020)

### **2.3 Platform-as-a-Service (PaaS)**

PaaS, or Platform-as-a-Service, is a distributed computing model that gives clients a total platform-hardware, programming, and framework for creating, running, and overseeing applications without the cost, multifaceted nature, and resoluteness of building and maintaining that platform on-premises. PaaS structures keep the fundamental framework far out of developers and different clients (IBM, 2020). The PaaS service supplier hosts everything, servers, storage, working framework programming, and information bases at their data centre. The client utilizes it just for a month to month charge dependent on utilization and can buy more assets on-request. The advantage of using PaaS is that it allows the user to build, deploy, update, maintain, test and scale the applications much more quickly and a lot cheaper than using their own platform to build and maintain the web apps or websites. (Violino 2019)

The unique feature of the PaaS administration compared to others is that it is the entire foundation platform rather than a specific application. The platform service provider organizations are liable for building up and keeping up this platform and conveying it to the application framework suppliers through administrations. Furthermore, PaaS suppliers likewise uphold specialized administrations, for example, application framework advancement and enhancement. The technical support team from PaaS suppliers additionally encourage the cycle of innovative work in a new application framework to help with the steady and long-haul activity. (Watts & Raza 2019)

The substance of PaaS is to change Internet assets administrations into a programmable interface, supporting outsider designers with business significant assets and platform. With the assistance of PaaS, designers can procure enormous programmable components that have an explicit business rationale to help advancement. This builds productivity besides it spares the expense. Web applications could be created easily and react to clients more quickly, which will carry strong advantages to the end clients consequently (Rouse 2020). PaaS has a unique service called middleware. Middleware is a program that sits in the middle of client confronting applications and the machine's working framework; for instance, middleware is the software that permits a program to get input from the console and mouse. The example of PaaS is Amazon Web Services (AWS), Google, Microsoft, IBM, Heroku and so on. (Cloudflare Inc, 2020)

## **2.4 Cloud application deployment**

Cloud application deployment refers to the enabling the end-users' or clients' access to SaaS (Software as a Service), PAAS (Service platform) or IaaS (Service Infrastructure). A model for cloud deployment refers to the type of cloud infrastructure on which a cloud solution is deployed. Cloud deployment requires the appropriate configuration and installation measures before a user supplying take place. (Atos SE, 2020)

An application built on the cloud is completely deployed and all aspects of the application are cloud-based. Cloud apps are either built on the cloud or transferred to cloud computing from existing infrastructure. Cloud-based systems can be created from low-level infrastructure components or can use a higher-level resource to meet core infrastructure specifications for management, design and Scaling. (AWS, 2020)

A cloud deployment model is defined according to the position of the deployment infrastructure and its power. Deciding what model to use is one of the most critical decisions users make on cloud deployment. Every model of cloud deployment meets various organizational requirements, so the user must select a model that meets user's organization needs. Maybe even more important is that there is a different value proposition and related costs for each model of cloud deployment. There are four types of cloud deployment models and they are private, public, community and hybrid cloud. (Rountree & Castrillo 2014)

### **3 HEROKU CLOUD PLATFORM**

Heroku is a cloud platform for organizations to develop, build, deliver, monitor, and scale applications. It is a cloud P-a-a-S (Platform as a Service). It enables developers to build, run, and operate applications on a cloud platform. Heroku was established in 2007. Heroku is the title of the item as well as the name of the company behind itself, and the company is called Heroku incorporated. Heroku is a Salesforce Incorporated affiliate and part of their cloud product catalogue called Salesforce App cloud. (Heroku Inc, 2020a)

According to Heroku's documentation, Heroku is a platform as a service that supports numerous programming languages. When Heroku was created in 2007 it only supported Ruby but now it supports programming languages like Java, Node.js, Scala, Python, Clojure, PHP, and Go. Heroku is a cloud application development platform that permits its clients to make and oversee application without any distraction of hardware and servers. Heroku accomplishes users by providing a platform to assist with application arrangements, configuration, management, and scaling. Heroku moreover offers other administrations such as data persistence tools. (Heroku Inc, 2020a)

Heroku is considered easy to use. Heroku is most beneficial for companies in specific circumstances. For small ventures, Heroku has a free service model, but for cases where more specific business requirements need to be met, tiered service packages exist. The Heroku cloud service platform is built on a container architecture with integrated data services and a powerful ecosystem (called dynos inside the Heroku paradigm) for deploying and running modern applications. Applications running on Heroku usually have a unique domain name used to redirect HTTP requests to the right container. (Rusev 2018)

#### **3.1 Heroku's terminology**

Heroku utilizes its arrangement of terms to impart different deliberations and ideas on the platform and redefine terms that have a changed and progressively explicit importance. Heroku looked at how they may be comprehended in various settings. The terms are recorded in sequential order in Table 1. (Heroku Inc, 2020c)

Feature	Definition
Add-on	Heroku add-on items are components that help your application such as data storage, information stockpiling investigation, information handling, These are entirely kept up for you by either an outsider supplier or by Heroku.
App	Applications comprised of your source code, a depiction of any conditions, and a Procfile
Build-pack	Build-packs lie behind the slug aggregation process. Build-packs take your application, its conditions, and the language runtime, and produce slugs.
Config-Var	Config vars (for example, arrangement factors) are seen by the application as an environmental variable and are utilized to design the app.
Dyno	Dyno is isolated and a virtualized container that helps to run the Heroku app's code.
Dyno Formation	A formation of how many dynos are currently executed for each process.
Free dyno	Using it, a developer can host a basic free Heroku app. It also has time restrictions.
Heroku CLI	The command-line interface which reacts with the Heroku's apps.
Heroku Dashboard	An interactive web interface that interacts with Heroku's apps.
Log drain	A Destination (i.e., a URL) that the Logplex administration courses the entirety of a Heroku application's logs to.
Logplex	A Heroku's Log delivery system that combines all the logs from various routes and shows them in a single application.
Maintenance Mode	A feature of Heroku's app, after enabling it helps to prevent all external web traffic from reaching.
One-Off Dyno	The transitory dynos that are run in their info/yield joined to clients' neighborhood terminal and to work on the Heroku application.
Pipeline	An assortment of Heroku applications that share the equivalent codebase, normally to speak to an application's turn of events, organizing, and creative adaptations.

Preboot	Heroku Preboot, which guarantees new web dynos are begun before existing ones are ended, has been changed to represent custom boot break settings.
Process type	Is declared by the Heroku app to indicate which command a dyno should run for a startup. Each dyno belongs to a specific process type, and it should use the particular command.
Procfile	Procfile is a document created by the client in the root catalogue of the Git storehouse containing the hotspot for the application that rundown orders to be executed by Heroku after the arrangement.
Region	A geographical location where an application dyno can be deployed.
Release	A slug joined with a lot of config vars is known as a release.
Release Phase	The occurrence of an optional phase just before the new app release is deployed.
Rollback	The path toward restoring a Heroku application to the state of a past release.
Slug	A slug is a heap of your source [code], fetched dependencies, the language runtime, and incorporated/created yield of the assemble framework - prepared for execution.
Stack	Heroku curates an operating system that runs on a dyno and system image.
Twelve Factor-app	A system for building programming as an administration application in present-day arrangement situations.
Web Dyno	A dyno can get HTTP traffic. These dynos run the order related to the web procedure type in an application's Procfile.
Worker Dyno	A dyno that can't get HTTP traffic. These dynos run the order related to any procedure type in an application's Procfile aside from the web procedure type.

TABLE 1. Heroku's Terminology (Ansaharju 2016)

## **3.2 Platform and solution stack**

The Heroku architecture is a platform stack that includes a language runtime, different libraries, the OS and underlying infrastructure to enable the creation of scalable web apps. According to the Orlando division of platform and solution stack, Heroku's characteristic is described. (Heroku Inc, 2020b)

### **3.2.1 Stack and build packs**

A platform where the Heroku application run is called a stack. A stack is an operating system picture that is curated and kept up by Heroku. Stacks are commonly founded on a current open-source Linux dis-semination, for example, Ubuntu. All the Heroku apps that are deployed now are using the latest stack 18.0 and its customized version is Ubuntu 18.04 operating system. On the head of the stack, speaking to the most reduced degree of the arrangement stack, Heroku offers manufacture conditions that are liable for transforming an application into a substance that is executable on the platform. These manufacture conditions are called build packs. Build packs come in various models from Heroku and can be made by the developers themselves too. Build packs are controlled by the slug compiler to deliver executable applications (slugs). Designers may make their build packs to make Heroku uphold extra language of programming structures. Heroku underpins running applications written in Ruby, Node.js, Java, Python, Clojure, Scala, Go and PHP programming languages by utilizing Heroku's legitimate build packs. A language runtime is given as a major aspect of the deployment stack. (Ansaharju 2016)

### **3.2.2 Procfile, process type and dynos**

After the deployment, an application may begin cycles of various kinds on the stage. The cycles are characterized in a Procfile in the application source code. The cycles are assigned figuring assets as Heroku's virtual processing units, dynos. Dynos come in different designs of figuring power, vaporous filesystem storage size and accessible memory, and can be scaled in types and numbers by cycle to make a dyno arrangement for an application. Dynos are normally appended to an application for its entire life aside from uncommon irregular dynos that can be utilized to perform non-tireless tasks. Dyno runtime consequently oversees dynos for an application, and dyno director deals with the dynos for all the platform. Figure 2 shows the complete architecture of Heroku. (Heroku Inc, 2020b)

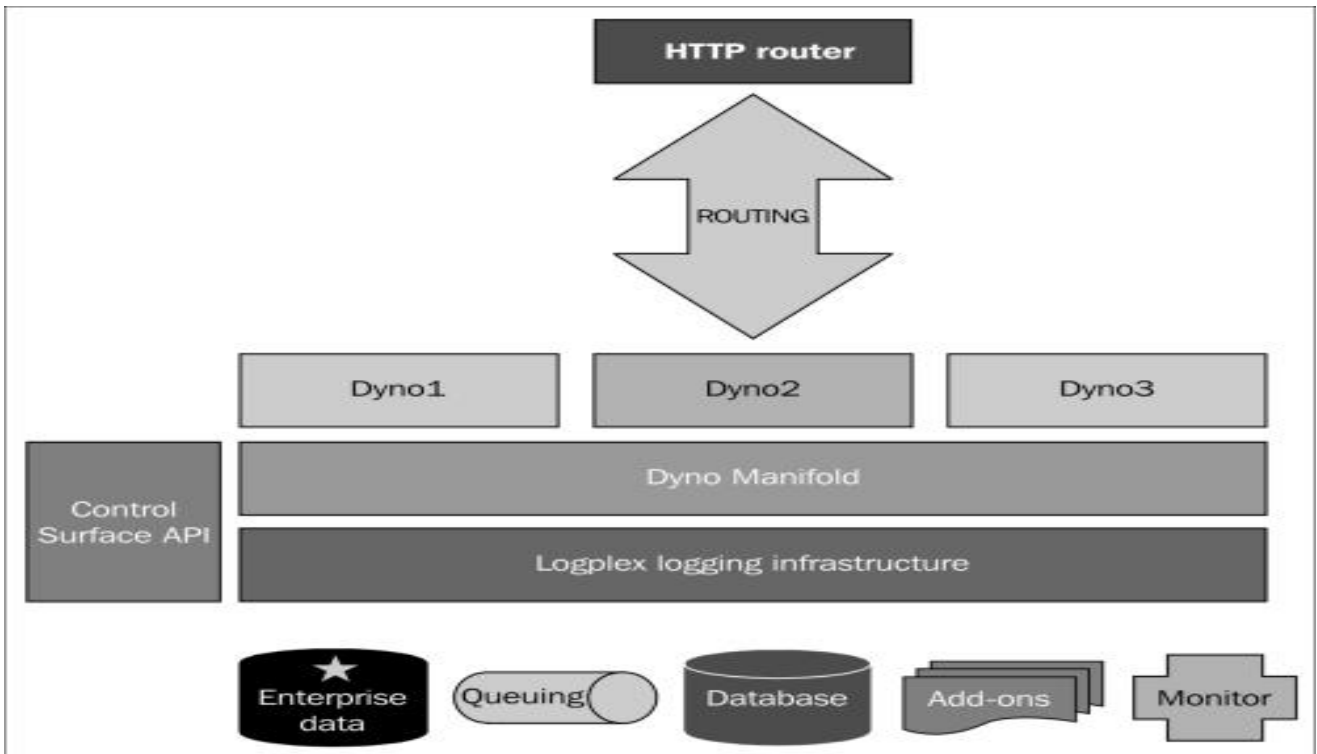


FIGURE 2. Heroku architecture. (Packt, 2019)

### 3.2.3 Routing, regions and add-ons

Heroku's application also includes modules that route network applications to (HTTP router) applications and aggregate logs from the log outputs for application and platform components. Applications of Heroku may be configured in a particular area of the world for the data center. The solution stack in Heroku is expanded by the ecosystem add-on. This allows third-party SaaS or PaaS providers to provide their services to software designers and application developers to incorporate the features of third parties into their applications easily. (Heroku Inc, 2020b)

### 3.3 Deployment flow

Heroku Flow integrates close integration with GitHub, visually presented pipelines, and device feedback with Heroku. Heroku Flow is designed to streamline the app release experience by making ongoing delivery quick, transparent, and effective. The Heroku Flow is a standardized deployment workflow. To make an application running in Heroku it should be deployed in Heroku. The application deployment process needs a certain task to perform. Firstly, an application source code, dependency declarations and

Procfile are created using a Git repository. The heroku app is created and is organized under the developer's Heroku user account either with the web interface of the Heroku dashboard or with the CLI tool. Heroku automatically allocates an application a Git repository endpoint. It is added as a remote store to the Git archive containing the application source by the developer. After the application source is transferred to the remote repository provided by Heroku using Git, the application will deploy to Heroku. Heroku will manufacture the application to a slug, make a release, allocate default dyno development to it and execute the procedure of type web pronounced in the Procfile. After the completion of all the processes, by default, an application is given to free dyno to sleep and save processing power. (Heroku Inc, 2020d)

### **3.4 Runtime management**


Runtime management is a part of the Heroku platform liable for running and dealing with an application. It also manages the app. The Heroku Runtime is responsible for provisioning and organizing compartments (dynos), overseeing, and checking their lifecycle, giving legitimate system arrangement, HTTP routing and log collection. After the deployment of an application user can scale applications on a level plane by changing dyno types and allocating more or fewer dynos per process. The user has the right to assign add-ons to the application and is able to set up an automatic deployment integration with Git and Dropbox. The user can manage collaborator access right and can execute the basic application management. Applications can be handled using a dashboard or a CLI tool through a web browser. (Heroku Inc, 2020e)

## 4 ARCHITECTURE OF A WEBSITE

A website is an essential asset for a company to do their business. It is the easiest way to reach out to their clients. Merely putting a dynamic website on the world wide web (WWW), a business owner can make the business available all around the world. The use of websites has been the reason for the web development industry has evolving rapidly during the last two decades. There is specialization in terms of web development, and they are Front-end development, Back-end development, and Full-Stack development. Even website has different parts and to make a complete website, we need all, but this thesis is focused on Front-end web development, which uses languages like HTML, CSS, and JavaScript. (Nguyen 2020)

### 4.1 HTML 5

HTML stands for Hypertext Markup language. Hypertext is a text that contains pieces of information by an electronic connection that allows users to access it using links. A markup language is a text that annotates electronic documents which is human readable. The markup language can be structured and formatted by computers. HTML is the standard markup language for building websites and web apps. HTML is represented by tags (<>). They allow users to create a section, division, container, headings, paragraphs, links to the web pages, and web apps. It is used to describe the structure of the webpages. As HTML is not a programming language, it is unable to create a functional and dynamic website. (Tech Terms, 2011)



```
<p>Hello World!</p>
```

FIGURE 3. HTML Code. (MDN, 2020).

The above Figure 3 is a simple example of how the HTML is used. *p* inside the tags represents the paragraph, and it ends with a closing tab. When you open an HTML file on a browser, then you can see a paragraph displaying Hello World! as seen in Figure 4 below.

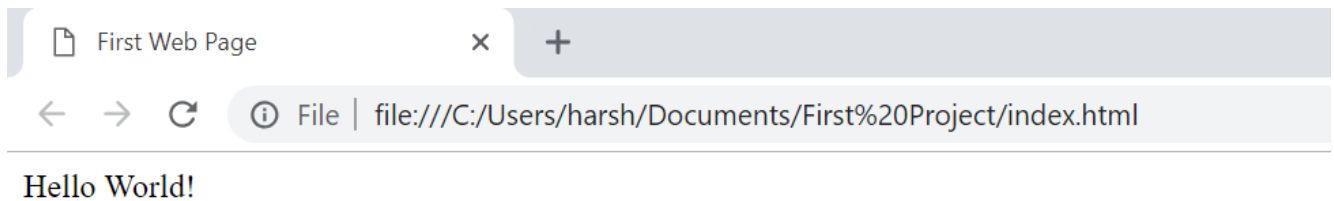


FIGURE 4. Hello World! (GeeksforGeeks, 2019)

HTML 5 is the latest version of HTML. The newest version of the HTML has unique attributes and elements. It has a large set of unique features and technologies that will help developers to develop more unique, diverse, and dynamic webpages and web apps. It also has JavaScript APIs for the improvement of storage and hardware access. It has over twenty new elements. The technologies are classified into several groups they are Semantic, Connectivity, Offline, and Storage, Multimedia, 2D and 3D effects, Performance and Integration, Device access, and Styling. (MDN, 2019)

## 4.2 CSS

CSS stands for Cascading style sheets and is a style sheet language. CSS is used to change the visual appearance of the HTML content (MDN, 2020). CSS describes how an element should be seen on the screen. CSS was first introduced in 1996. CSS has become the base of modern website design and development. There are three ways to include CSS into HTML, and they are inline, internal, and external styling. Firstly, Inline CSS allows a developer to apply a unique style in one HTML element. Inline CSS is used to style the CSS element quickly. This styling is rarely used and used for small webpages. The best advantage of inline styling is that it will override the internal and external CSS. For instance, if a

developer wants to change the style of one unique HTML element without changing other styles, this will be the best way to do so. The example of inline CSS is shown below in Figure 5. (Amy 2011).

```
<p style="color: blue;">This is a paragraph.</p>
```

FIGURE 5. Inline CSS. (MDN, 2020c).

Another method of incorporating CSS in HTML is internal CSS. Mainly, it is used to define the stylesheet of a single HTML webpage. It is included in the main HTML page inside the *<head>* section by incorporating all the style inside the *<style>* tag. It is the most effective way to style a single page HTML file. The advantage of using internal CSS is, a developer can use class and id to style the web page. The example of inline CSS is shown below in Figure 6.

```
<head>
  <style>
    body{
      background-color: red;
    }
  </style>
</head>
```

FIGURE 6. Internal CSS. (MDN, 2020c).

The last method to incorporate CSS in HTML is external CSS. It is the most effective way to style the website with multiple pages. All the styles can be created in one single file and later linked with the HTML file. It is a style in a separate file and stores all the CSS styles in it. The external file is also included in the head section of the HTML as in internal, but the only difference is, it is added using a link. The example of linking external CSS file is given below in Figure 7.

```
<head>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
```

FIGURE 7. External CSS. (MDN, 2020c).

Apart from giving their style to a webpage, developers also have an option to use CSS libraries and frameworks. They are the ready-made stylesheets, which are easy to use. These libraries will be time saving for the developers. To do a big project with a lot of styling needed, these libraries will be an essential tool for web developers (Rachel 2018). As ready-made design cannot satisfy or meet the developers demand, there is also an option for the developers if they don't want to use the whole stylesheet, they can override the style. Besides, the framework CSS has pre-processors like SASS (Syntactically Awesome Styles Sheets), which has the most mature, stable, and unique professional grade CSS. These pre-processors allow developers to use variables, functions, nested rules, and more with all CSS-compatibility syntax. (w3schools, 2020)

### **4.3 Bootstrap**

Bootstrap is the most well-known HTML, CSS, and JS framework to build up a responsive site. The ex-Twitter employees Mark Otto and Jacob Thronton created it. Bootstrap incorporates HTML and CSS based configuration layouts for familiar UI. The designs incorporate components like buttons, drop-downs, typography, tabs, structures and tables, and so on, and it moreover includes discretionary JavaScript extensions. As indicated in layman's terms – “Bootstrap is a handy combination of supportive, reusable bits of code written in HTML, CSS, and JavaScript” (Ouellette 2017). It's likewise a front-end development system that empowers developers and designers to develop completely responsive sites rapidly. Bootstrap 4.5 is the latest adaptation of Bootstrap. (Ouellette 2017)

The distinction of Bootstrap has to do with the features it has. The speed of development is one of the key features it has. It is progressively useful concerning designing an application or site in a short period. Bootstrap enables the use of a block of codes to kick the arrangement off. There is no prerequisite for designing without any planning, and the developers can use the block of codes to design any part of the site. A considerable lot of website designing and developing organizations are utilizing this way to deal with conveying the product to their clients in a short period. (Ouellette 2017)

As Bootstrap 4.5 was the latest version of Bootstrap when this research was done, it has updated the features and there are new interaction utilities, new reboot style for pointer cursor, examples of Bootstrap are downloadable, save up to ~5% for the compressed minified JS builds and the most important one is they are added to docs on how to use and install Bootstrap. There is an update of the JQuery and Jekyll to version 3.5.1 and 4 respectively. (Bootstrap, 2020)

There are different methods to install Bootstrap to the project. Firstly, users have access to download the compiled CSS and JS files. Then a user can include the downloaded file in the project and have all the entrance to the source code. Another way is to install it through the package manager like npm, RubyGems, and more. The example is shown below in Figure 8 and 9 respectively.

```
$ npm install bootstrap
```

FIGURE 8. Installing Bootstrap using the npm package manager. (Bootstrap, 2020).

```
$ gem install bootstrap -v 4.5.0
```

FIGURE 9. Installing Bootstrap using a gem package manager. (Bootstrap, 2020).

Apart from downloading the file, users also have the option to use the bootstrap CDN (Content Delivery Network) in which users just have to copy the link of bootstrap CDN to the head section of the main Html page. Using this CDN user has the cached version of Bootstrap with compiled CSS and JS. The example of using CDN is presented below in th Figure 10.

```
<head>  
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" >  
</head>
```

FIGURE 10. Bootstrap CDN. (Bootstarp, 2020).

After using the JS compiled Bootstrap CDN user should not forget to use the jQuery without including the jQuery CDN. It will not work, or JS will not be functional. The jQuery CDN should be incorporated into the body section of the main Html page. The example is presented below in Figure 11. (Bootstrap 2020)

```
<body>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js" ></script>
  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" ></script>
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" ></script>
</body>
```

FIGURE 11. JS compiled Bootstrap CDN. (Bootstarp, 2020).

#### 4.4 Javascript

JavaScript is known as a lightweight, high-level compiled programming language. It was founded by Netscape Communications' Brendan Eich in 1995. Netscape Communications was trying to build a scripting language that not only augmented Java but also had a syntax, except languages like Perl, Python and TCL, which were accessible at the time. The standard of JavaScript is ECMAScript (ECMA stands for the Association of European Computer Producers) (Nguyen 2020). JavaScript is considered as both front end and back end web developing language. JavaScript is simple to learn and a compelling scripting language, broadly utilized for controlling web page behaviour. JavaScript's recently released version is ES6. (Javascript.info, 2020)

JavaScript is used to render web pages more complex. In essence, what web pages do with HTML or CSS is to show details. Web pages can perform more active activities with JavaScript, such as timely updating content, animations, interactive maps, etc. One of the essential features of JavaScript is to send requests and receive the data from the server using APIs. The API has the same effect as a protocol of communication among devices (browser, computer, server, etc.). This function is the biggest difference between a static web page and a dynamic web page because it is hard coded in the content of a static web page, while the dynamic web page gets modified content from servers. (Code-Club, 2020)

Developers may create a website or application from scratch, but this might not be necessary often. There are common characteristics that are used frequently. In this instance, JavaScript libraries offer ready-to-use solutions to save time by building a website. jQuery, which was made in 2006, is one of the first libraries. jQuery is known for selecting elements and their fields. However, developers can do the same thing recently with just pure, newer JavaScript. In that case, jQuery is not needed, unless developers choose to use other functions. (Nguyen 2020)

## 5 IMPLEMENTATION

Website development is a very long process that includes a variety of activities, resources, and materials. A comprehensive plan helps to create websites in a particular time and budget. To make a website development process simpler, it requires a good project plan. It is done one step at a time in various phases. The various stages in the website development process include the description of website goals and objectives, the grouping of a content list, the concept development, wireframe, the navigation framework and the implementation of the design and the publication finally. Each step avoids confusion and allows the development process to run smoothly. (Dummies 2017)

### 5.1 Website wireframing

A wireframe is a normal outline of a project. It is the initial stage of starting any kind of project. The process of making a wireframe is called wireframing. A wireframe of a web page is a mock-up of how the webpage is going to look up after it is published. It helps a developer to make sure that all the sections and features are in the right place of a website. Wireframes are normally made after the webpage guide or flowchart before structuring a website. Wireframing is a significant part in website improvement measure as it fills in as a plan or sketch of the site. Building a site without making a wireframe resembles building a house without a plan. Wireframes can set aside time and cash in any venture. With wireframing, the entire advancement cycle can be run easily as the engineer has clear thought of what they are building. (Edward 2020)

To sketch a wireframe there are a lot of applications and software like Smart Draw, OmniGraffle, Microsoft Vision but in this project, draw.io is used as a tool to create the wireframe. Though the most simple way to draw a wireframe is using a pen and a pencil so, a good wireframe and guidelines help designers and developers a lot. The wireframing of the website is presented below in Figure 12. (Pazos 2018)

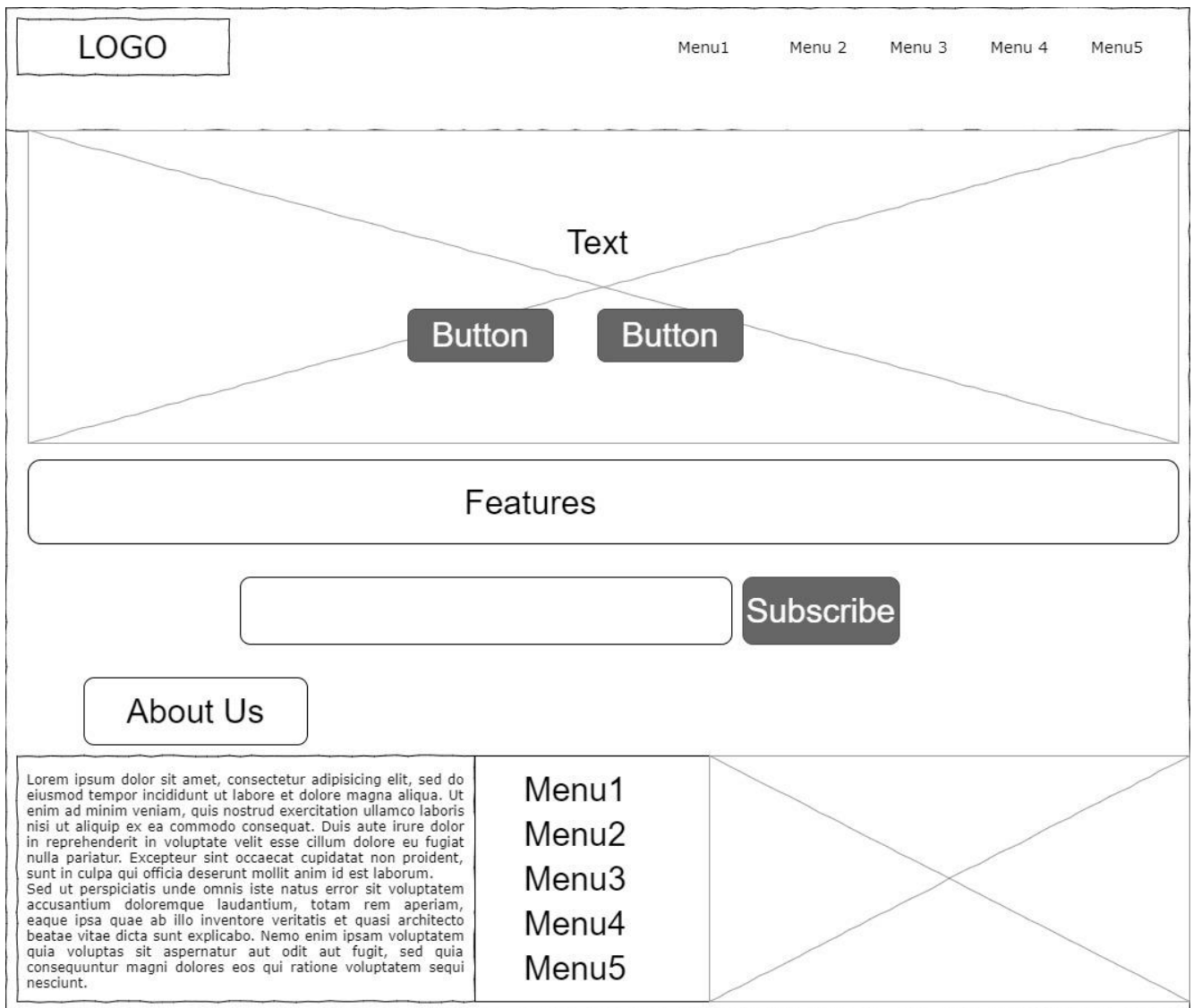


FIGURE 12. Website wireframing.

## 5.2 Website structure

A website is a collection of different web pages. So, to create a good website, webpages must be well-structured. A good-looking website is important but at the same time how quickly it responds also attracts the user. For the quick response of the website, web structure plays a crucial role. For example, a well-structured mall enables customers to find stores quickly, likewise a well-structured website makes users go through the webpages easily and quickly. This also makes a website user-friendly. So, to make a website structure good a detailed plan is necessary. Arranging a framework of the websites' structure before beginning to develop the webpages is fundamental because a framework of the website's structure fills in as a diagram of the website. Apart from this how the clients want to build a website is also

important. Above all the webpages are created in such a way that they will link with the homepage. Figure 13 shows the website structure. (Hitesh 2020)

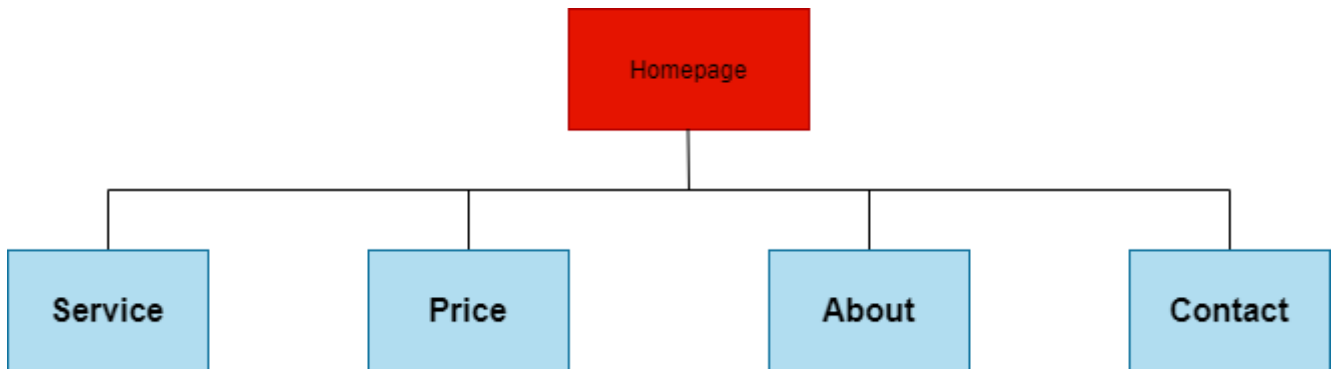


FIGURE13. Website Structure.

### 5.3 Development

The first step for website development is to open the text editor and set up a project. It includes several files. The first file is the **index.html** file which is also the main page of the website. Another file is the **styles.css** which is the external CSS file where all the styles are given and later linked it in the main **index.html** file. In the same way, an external file of JavaScript is also included called **index.js**. Figure 14 shows the files of the project.

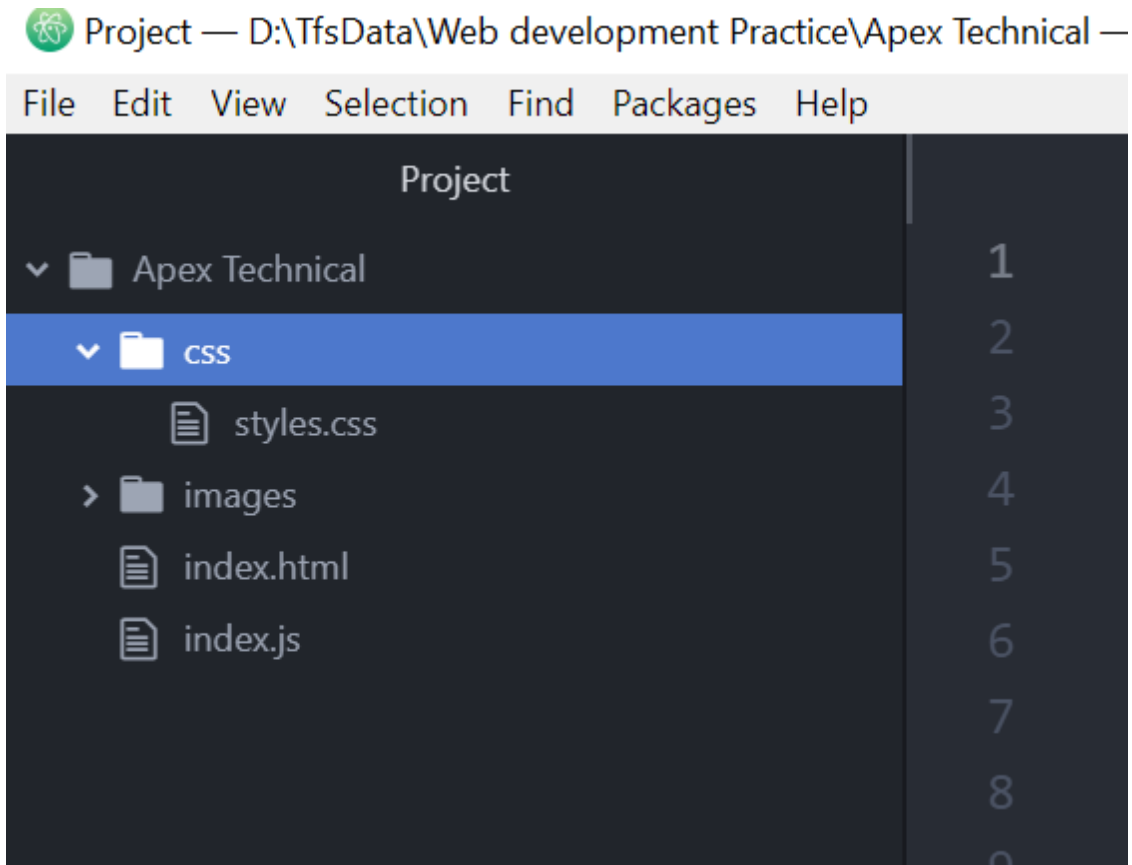
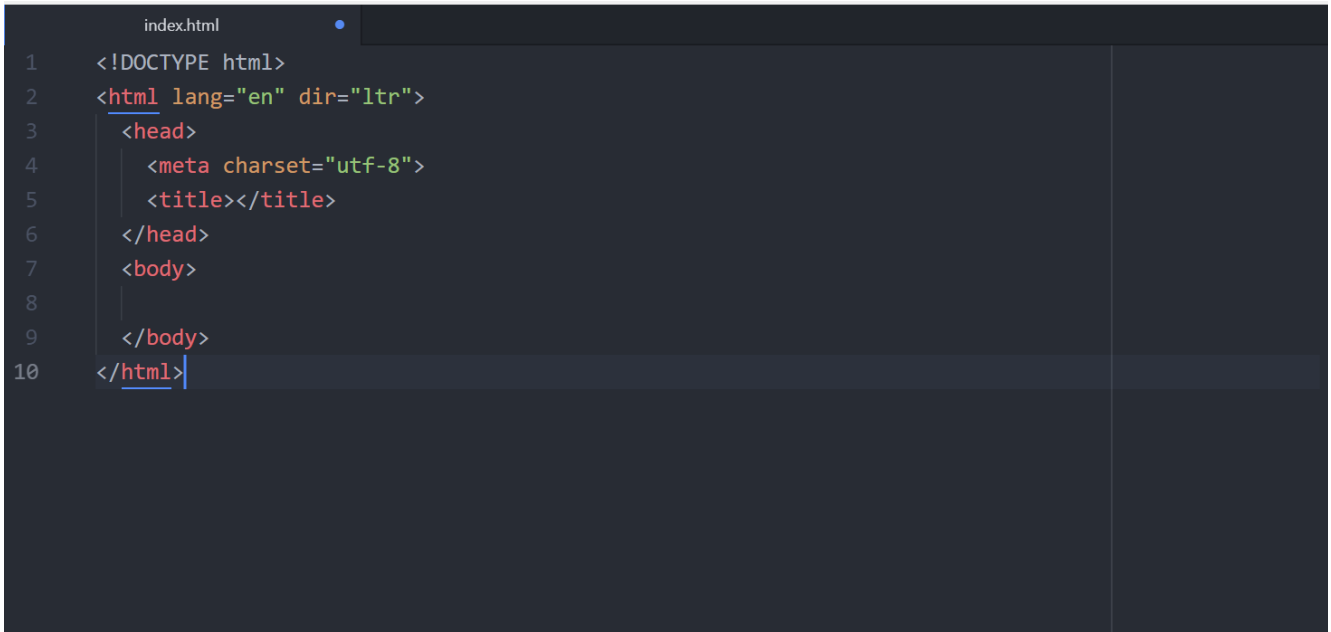


FIGURE 14. Setting up the project.

As a project folder is set, A bunch of source code is written to get the structure of the website. The first step is to write code in the index file of the website. To begin with, all the HTML tag and code are written in the `index.html` page to give the structure of the website. Figure 15 shows the HTML boilerplate.

A screenshot of a code editor window titled 'index.html'. The editor shows a standard HTML boilerplate structure. The code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en" dir="ltr">
3   <head>
4     <meta charset="utf-8">
5     <title></title>
6   </head>
7   <body>
8
9   </body>
10 </html>
```

The code is color-coded: <!DOCTYPE html> is blue, <html> is green, <head> is red, <meta> is green, <title> is red, </head> is red, <body> is red, </body> is red, and </html> is green. The cursor is positioned at the end of the </html> tag on line 10.

FIGURE 15. HTML boilerplate.

Depending upon the text editor the user can get this boilerplate. As the atom is used in this project, so simply writing HTML and tap enter key will show this boilerplate. They have their meaning as *<!DOCTYPE html>* meaning it is the declaration that it will only be HTML mark-up language in this document. The *<html>* opening and closing tag refer to that all the elements should be contained inside it. It is the core element. *<title>* refers to what title is given for a specific website whereas inside the *<body>* tag all the content of the page are inside it which is visible to the user. After starting with these all the codes are written in the code editor and also many images were used during the development. Figure 16 shows the code written in the index.file

```

67 <!-- -----Testimonials Starts----->
68 <section class="header-extradiiv">
69   <div class="container">
70     <div class="row">
71       <div class=" extra-div col-lg-4 col-md-4 col-12">
72         <a href="#"><i class="fas fa-3x fa-desktop" aria-hidden="true"></i></a>
73         <h2>EASY TO USE</h2>
74         <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor inci
75       </div>
76
77       <div class=" extra-div col-lg-4 col-md-4 col-12">
78         <a href="#"><i class="fas fa-3x fa-trophy" aria-hidden="true"></i></a>
79         <h2>AWESOME DESGIN</h2>
80         <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor inci
81       </div>
82
83       <div class=" extra-div col-lg-4 col-md-4 col-12">
84         <a href="#"><i class="fas fa-3x fa-magic" aria-hidden="true"></i></a>
85         <h2>EASY TO CUSTOMIZE</h2>
86         <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor inci
87       </div>
88
89

```

FIGURE 16. Index HTML file.

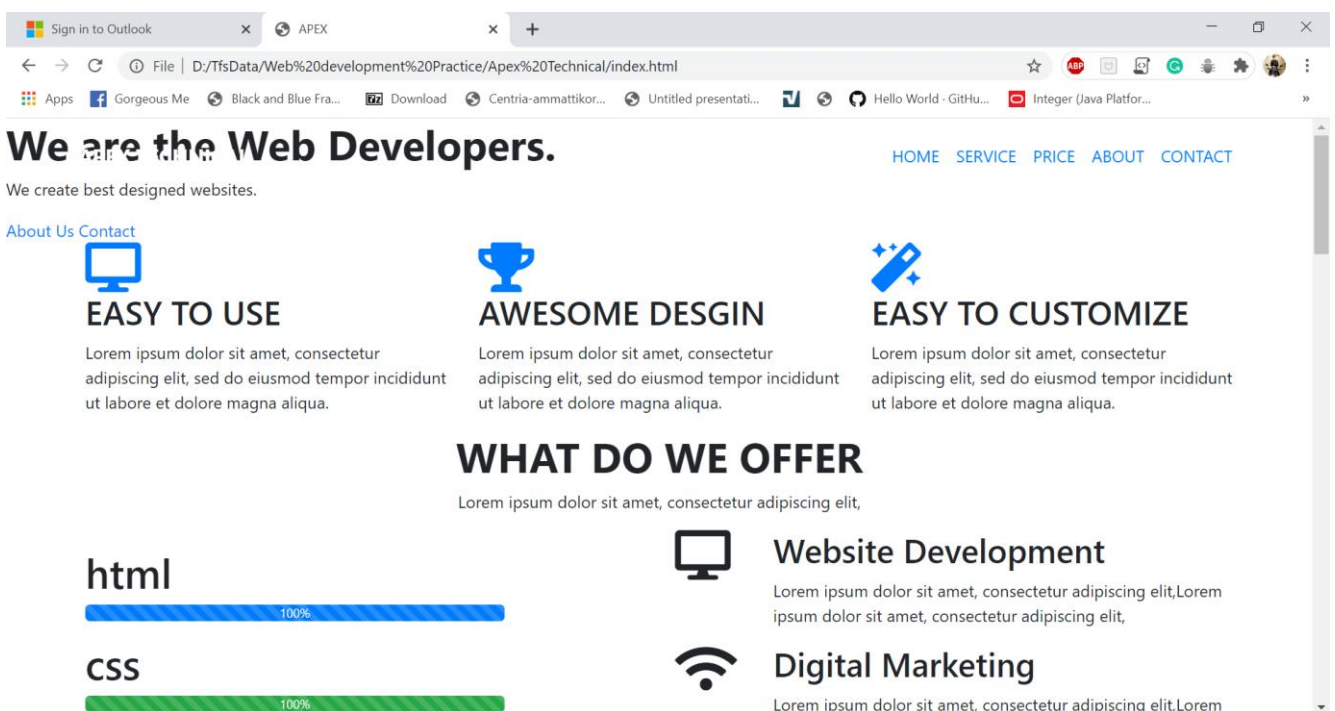
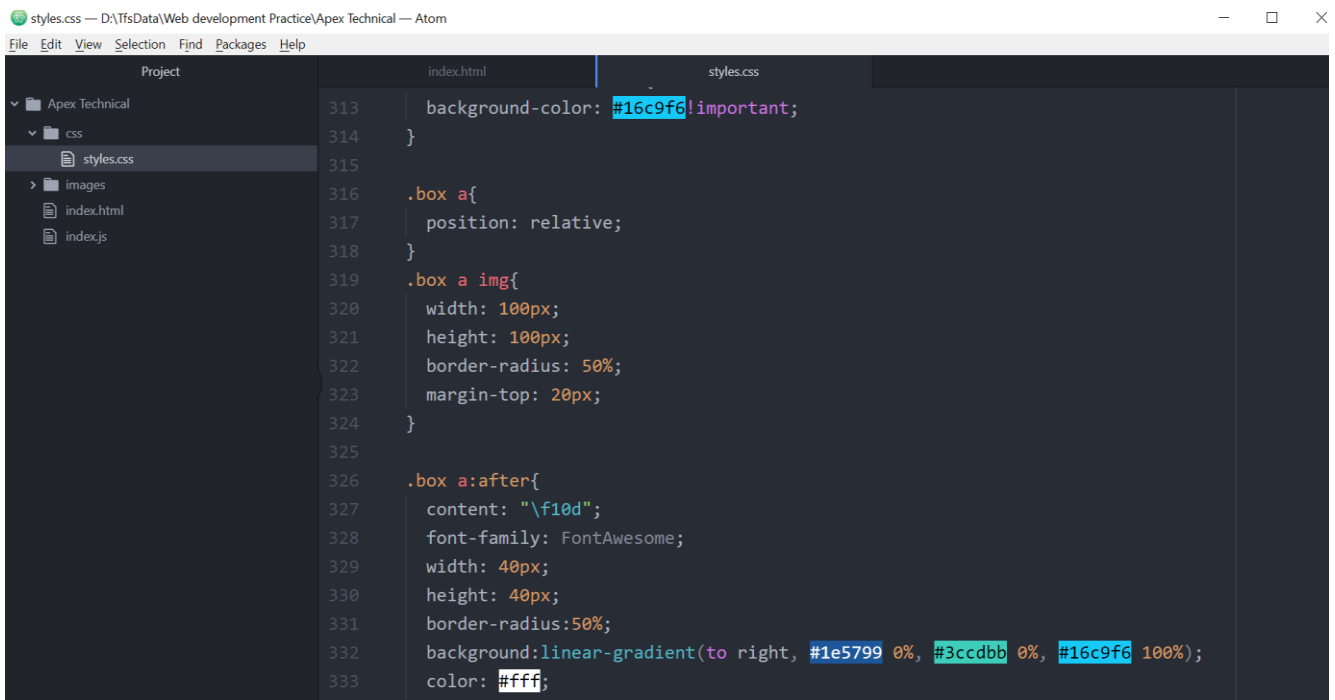


FIGURE 17. HTML file without CSS.

Figure 17 shows the HTML file of the website without any kind of styling, but it seems there is already some kind of styling added. The reason that the styles are seen in the normal HTML file is the default styles of Bootstrap. The Bootstrap has its default style. So before giving the style using CSS to each elements Bootstrap gives their style to those elements. After the HTML file is ready, CSS is added to

the website. Figure 18 below shows the CSS stylesheet. After giving styles to HTML file using CSS, it will override Bootstrap's default style. Figure 19 shows the HTML file after CSS is added which overrides the Bootstrap's style.



```

313 background-color: #16c9f6 !important;
314 }
315
316 .box a {
317   position: relative;
318 }
319 .box a img {
320   width: 100px;
321   height: 100px;
322   border-radius: 50%;
323   margin-top: 20px;
324 }
325
326 .box a:after {
327   content: "\f10d";
328   font-family: FontAwesome;
329   width: 40px;
330   height: 40px;
331   border-radius: 50%;
332   background: linear-gradient(to right, #1e5799 0%, #3ccdbb 0%, #16c9f6 100%);
333   color: #fff;

```

FIGURE 18. Adding CSS to HTML file.

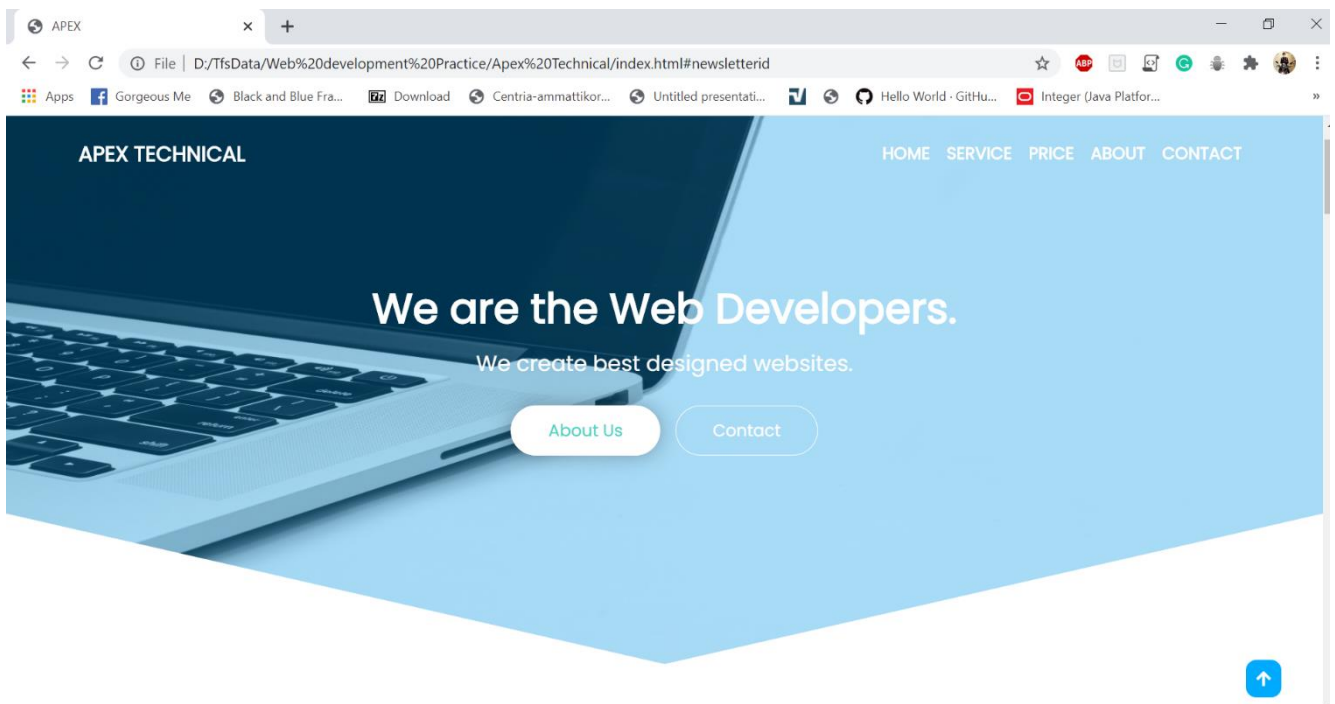


FIGURE 19. After adding CSS to HTML file.

As the styling of the whole website is completed, the next step is to make the navigations key work. As the website is a single-page website, it does not have multiple pages so there is no need of linking page paths. So, for the navigation, the id of each section is created to navigate by clicking the navigation bar menus. Figure 20 shows an example of how it is used.

```
<a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>
</li>
<li class="nav-item">
  <a class="nav-link" href="#servicediv">Service</a>
</li>
<li class="nav-item">
  <a class="nav-link" href="#pricingdiv">Price</a>
</li>
<li class="nav-item">
  <a class="nav-link" href="#newsletterid">About</a>
</li>
<li class="nav-item">
  <a class="nav-link" href="#contactid">Contact</a>
</li>
</ul>
```

FIGURE 20. Navigation using id.

Finally, JavaScript is added to the website to make it functional and interactive. A scroll-top button is created to take a user from the bottom to the top directly. A button is not seen at the top of the home page but when the user scrolls down it will appear. So, the user can just click the button to come to the top again. Figure 21 shows how JavaScript is used. After this, a complete website will be ready.

```

$('.count').counterUp({
  delay: 10,
  time: 4000
})

mybutton = document.getElementById("myBtn");

window.onscroll = function() {scrollFunction()};

function scrollFunction(){
  if(document.body.scrollTop > 20 || document.documentElement.scrollTop > 20){
    mybutton.style.display = "block";
  }else {
    mybutton.style.display = "none";
  }
}

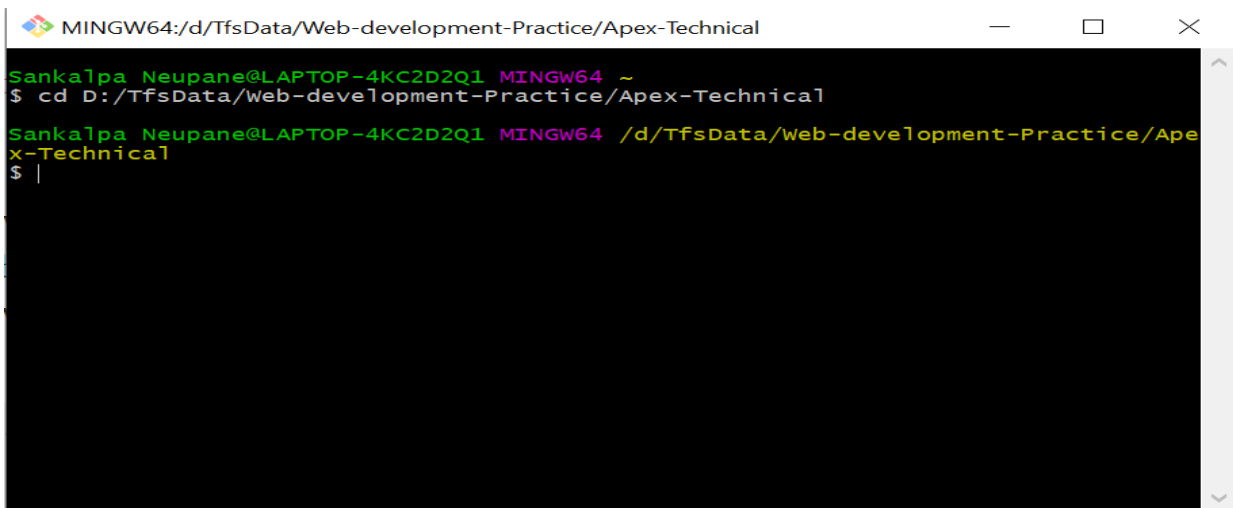
function topFunction() {
  document.body.scrollTop = 0;
  document.documentElement.scrollTop= 0;
}

```

FIGURE 21. JavaScript File.

## 5.4 Website deployment

A website deployment is to move a website from the local environment to live servers. In other word, deployment is the process of deploying the code from the source platform to a hosting platform. Even if it sounds very simple, it can be quite complex. There are several ways to deploy a website. (Chris 2015). After the website is created, it is ready to deploy. The website is deployed using a command so the first step to deploy a website to Heroku is to install Heroku CLI and Git bash terminal. After the installation, all the deployment process is done using the commands in Git bash terminal. There will be a default directory which is the user of the computer. Now the directory should be changed and navigate to the root directory of the static website. Figure 22 below presents the way to use Git bash terminal.



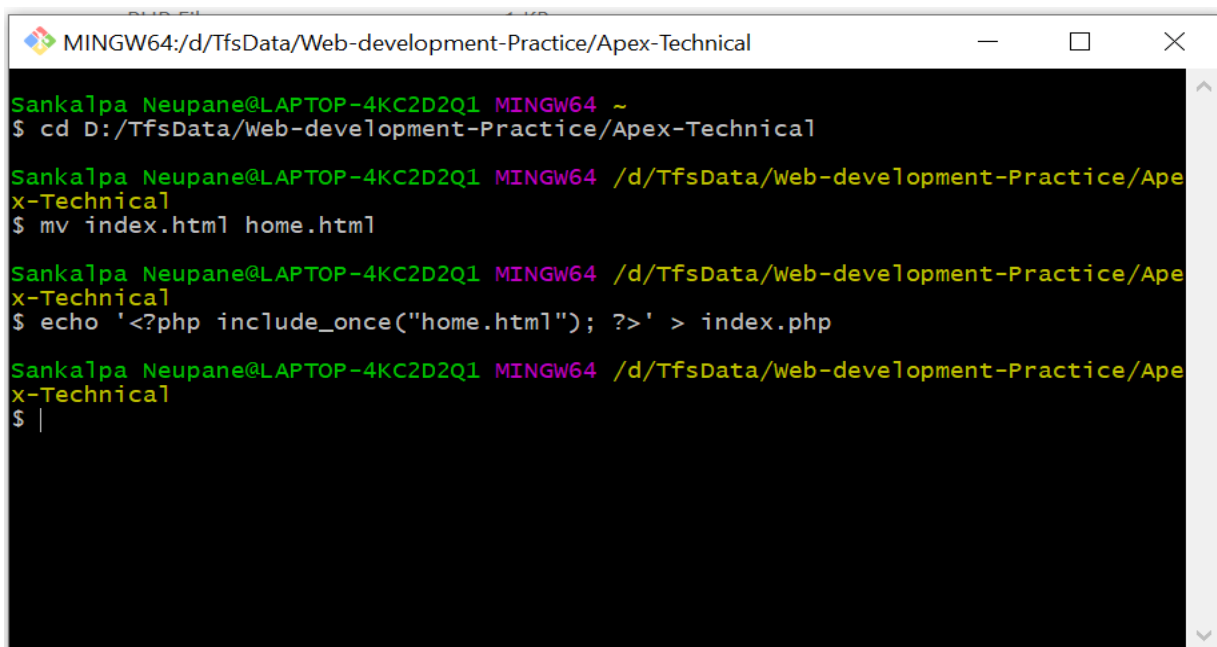
```

MINGW64:/d/TfsData/Web-development-Practice/Apex-Technical
Sanka1pa Neupane@LAPTOP-4KC2D2Q1 MINGW64 ~
$ cd D:/TfsData/Web-development-Practice/Apex-Technical
Sanka1pa Neupane@LAPTOP-4KC2D2Q1 MINGW64 /d/TfsData/Web-development-Practice/Ape
x-Technical
$ |

```

FIGURE 22. Git bash terminal and changing of the directory.

After changing the directory, in the main folder, the name of index.html is changed to home.html. A new single dynamic PHP file is added to the same folder called index.php. The reason behind including the PHP file is that Heroku supports PHP programming language. The index.php file includes the main home.html file, so the index.php file redirects to the static page. All the activities can be done using the terminal with some command and they are shown in the Figure 23 below.



```

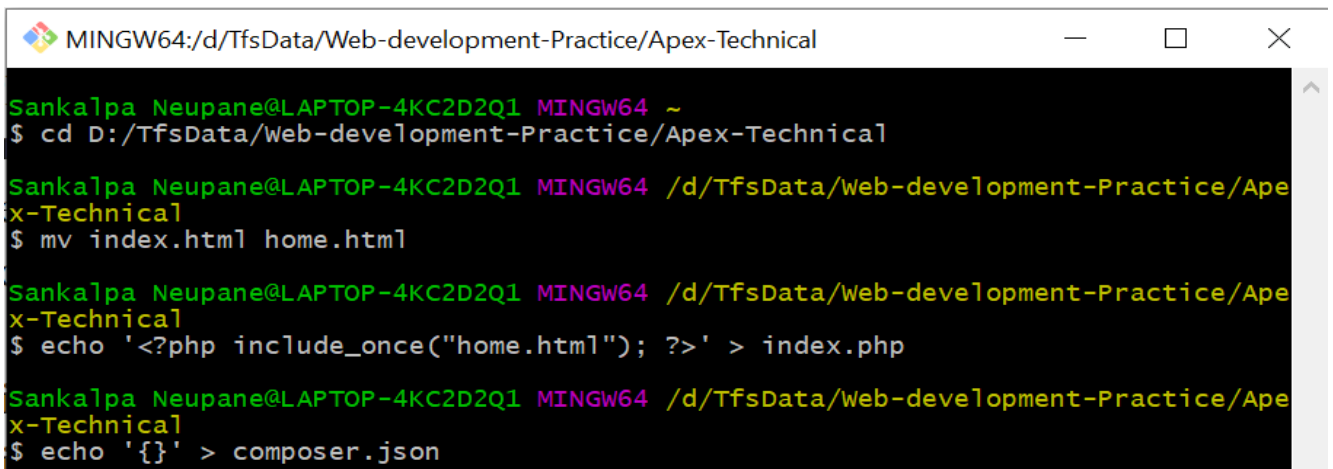
MINGW64:/d/TfsData/Web-development-Practice/Apex-Technical
Sanka1pa Neupane@LAPTOP-4KC2D2Q1 MINGW64 ~
$ cd D:/TfsData/Web-development-Practice/Apex-Technical
Sanka1pa Neupane@LAPTOP-4KC2D2Q1 MINGW64 /d/TfsData/Web-development-Practice/Ape
x-Technical
$ mv index.html home.html
Sanka1pa Neupane@LAPTOP-4KC2D2Q1 MINGW64 /d/TfsData/Web-development-Practice/Ape
x-Technical
$ echo '<?php include_once("home.html"); ?>' > index.php
Sanka1pa Neupane@LAPTOP-4KC2D2Q1 MINGW64 /d/TfsData/Web-development-Practice/Ape
x-Technical
$ |

```

FIGURE 23. PHP inclusion using commands.

Next, a composer.json file is also created in the same folder where index.php is created. Even though the JSON file is kept empty it will not make any impact but to get rid of any kind of warnings from Heroku normal curly brackets are added to the file this can be done using the terminal. The main reason

behind adding the JSON file is because the Heroku PHP supports will be applied only when a composer.json file is created in the root directory. Figure 24 shows how to create a JSON file. So, after the completion of creating the required files, we can proceed to the process of deployment.



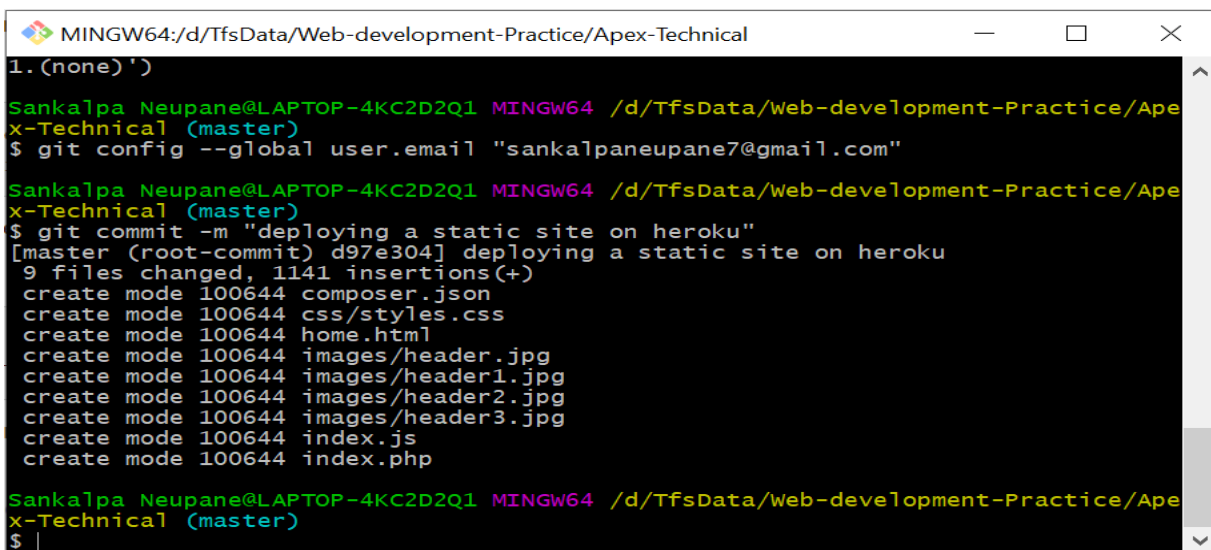
```

MINGW64:/d/TfsData/Web-development-Practice/Apex-Technical
Sankalpa Neupane@LAPTOP-4KC2D2Q1 MINGW64 ~
$ cd D:/TfsData/Web-development-Practice/Apex-Technical
Sankalpa Neupane@LAPTOP-4KC2D2Q1 MINGW64 /d/TfsData/Web-development-Practice/Apex-Technical
$ mv index.html home.html
Sankalpa Neupane@LAPTOP-4KC2D2Q1 MINGW64 /d/TfsData/Web-development-Practice/Apex-Technical
$ echo '<?php include_once("home.html"); ?>' > index.php
Sankalpa Neupane@LAPTOP-4KC2D2Q1 MINGW64 /d/TfsData/Web-development-Practice/Apex-Technical
$ echo '{} ' > composer.json

```

FIGURE 24. Command for creating a JSON file.

An app should be created to Heroku before the process of deployment. Next, the git is initialized, and all the files are added to the git repository using the terminal by giving commands called git init and git add respectively. Next, to save all the previous work to the git by committing a message what has been done till now. The main thing to keep in mind when committing this message, is that the git bash terminal should be previously signed in with the same id of GitHub or it should be updated in the terminal using the command. Otherwise, the commit will not be processed. Figure 25 shows how to add the git id and commit a message.



```

MINGW64:/d/TfsData/Web-development-Practice/Apex-Technical
1. (none)')
Sankalpa Neupane@LAPTOP-4KC2D2Q1 MINGW64 /d/TfsData/Web-development-Practice/Apex-Technical (master)
$ git config --global user.email "sankalpaneupane7@gmail.com"
Sankalpa Neupane@LAPTOP-4KC2D2Q1 MINGW64 /d/TfsData/Web-development-Practice/Apex-Technical (master)
$ git commit -m "deploying a static site on heroku"
[master (root-commit) d97e304] deploying a static site on heroku
9 files changed, 1141 insertions(+)
 create mode 100644 composer.json
 create mode 100644 css/styles.css
 create mode 100644 home.html
 create mode 100644 images/header.jpg
 create mode 100644 images/header1.jpg
 create mode 100644 images/header2.jpg
 create mode 100644 images/header3.jpg
 create mode 100644 index.js
 create mode 100644 index.php
Sankalpa Neupane@LAPTOP-4KC2D2Q1 MINGW64 /d/TfsData/Web-development-Practice/Apex-Technical (master)
$

```

FIGURE 25: Command for git id and commit message.

Now, the final stage is to move all the files to the Heroku. Heroku is logged in to move the files. After using the command, the terminal directly takes a user to the terminal login page of the Heroku. Clicking the login button will allow creating a web app in the Heroku. All the process stages are shown below in the figures 26, 27 and 28, respectively.

```
Sankalpa Neupane@LAPTOP-4KC2D2Q1 MINGW64 /d/TfsData/Web-development-Practice/Ape
x-Technical (master)
$ Heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/5c0fafe2-df6e-47
bd-bcb0-959145c57d9b?requestor=SFMyNTY.g3QAAAACZAAEZGF0YW0AAAAA00DQuMjQ4LjExNi4xM
jRkAAZzaWduZWwRuBgAhEjR4dAE.gspqavsOEMwQGULo06cr3amJUnwWduk1eKt5myZBVzo
heroku: waiting for login...
Logging in... done
Logged in as sankalpaneupane7@gmail.com
```

FIGURE 26. Heroku login.

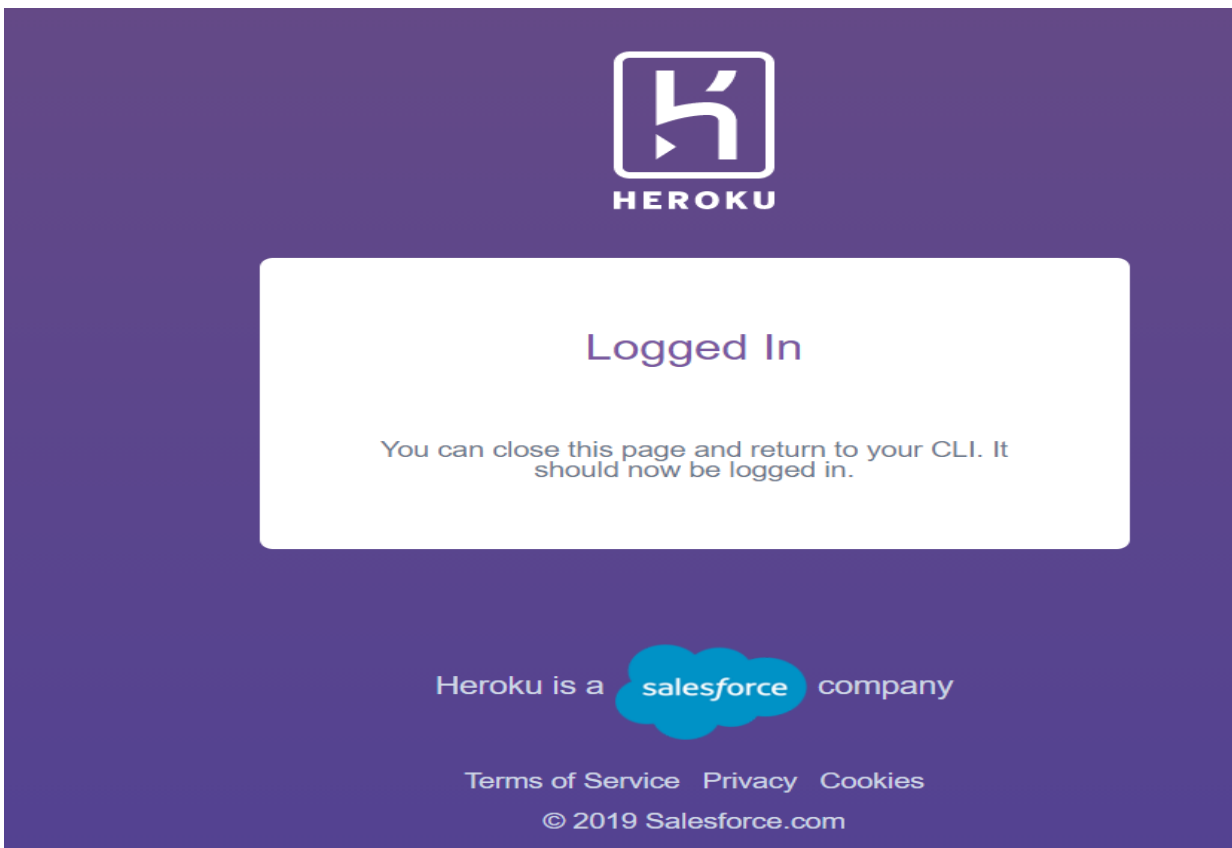


FIGURE 27. Login page of Heroku.

```
Sankalpa Neupane@LAPTOP-4KC2D2Q1 MINGW64 /d/TfsData/Web-development-Practice/Ape
x-Technical (master)
$ heroku apps:create apex-technical
Creating apex-technical... done
https://apex-technical.herokuapp.com/ | https://git.heroku.com/apex-technical.gi
t
```

FIGURE 28. A command for creating an app on Heroku.

The app is created in the Heroku. The final step is to deploy the static website. To deploy using the command mean all the files will be moved to Heroku, it will first detect the type of the application which is deployed, compile the app and make it accessible over the Internet. Figure 29 shows how to deploy the app created in Heroku. This below link is the example of the deployed app.

<https://apex-technical.herokuapp.com/#>

```
Sankalpa Neupane@LAPTOP-4KC2D2Q1 MINGW64 /d/TfsData/Web-development-Practice/Ape
x-Technical (master)
$ git push heroku master
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 4 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (13/13), 2.79 MiB | 1.52 MiB/s, done.
Total 13 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> PHP app detected
remote: -----> Bootstrapping...
remote: -----> Installing platform packages...
remote:      NOTICE: No runtime required in composer.lock; using PHP ^7.0.0
remote:      - php (7.4.9)
remote:      - apache (2.4.46)
remote:      - nginx (1.18.0)
remote: -----> Installing dependencies...
remote:      Composer version 1.10.10 2020-08-03 11:35:19
remote: -----> Preparing runtime environment...
remote:      NOTICE: No Procfile, using 'web: heroku-php-apache2'.
remote: -----> Checking for additional extensions to install...
remote: -----> Discovering process types
remote:      Procfile declares types -> web
remote:
remote: -----> Compressing...
remote:      Done: 16.9M
remote: -----> Launching...
remote:      Released v3
remote:      https://apex-technical.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/apex-technical.git
 * [new branch]      master -> master

Sankalpa Neupane@LAPTOP-4KC2D2Q1 MINGW64 /d/TfsData/Web-development-Practice/Ape
x-Technical (master)
$ |
```

FIGURE 29. Deployment process.

## 5.5 Running the app on Heroku

As the website deployment is deployed in Heroku it can be used from the server instead of the local environment. Heroku supports web application where it does not support static websites. So, deploying a static website in Heroku will be a challenge. In the case of deploying a web application the user can follow the normal way of deploying but in case of static webpages or websites, the user must follow several steps. Firstly, make sure that Git and Heroku CLI are installed on the PC along with these users

should have GitHub and Heroku accounts. Figure 30 and 31 represents the Heroku’s user dashboard and a web application running on the servers using Heroku respectively.

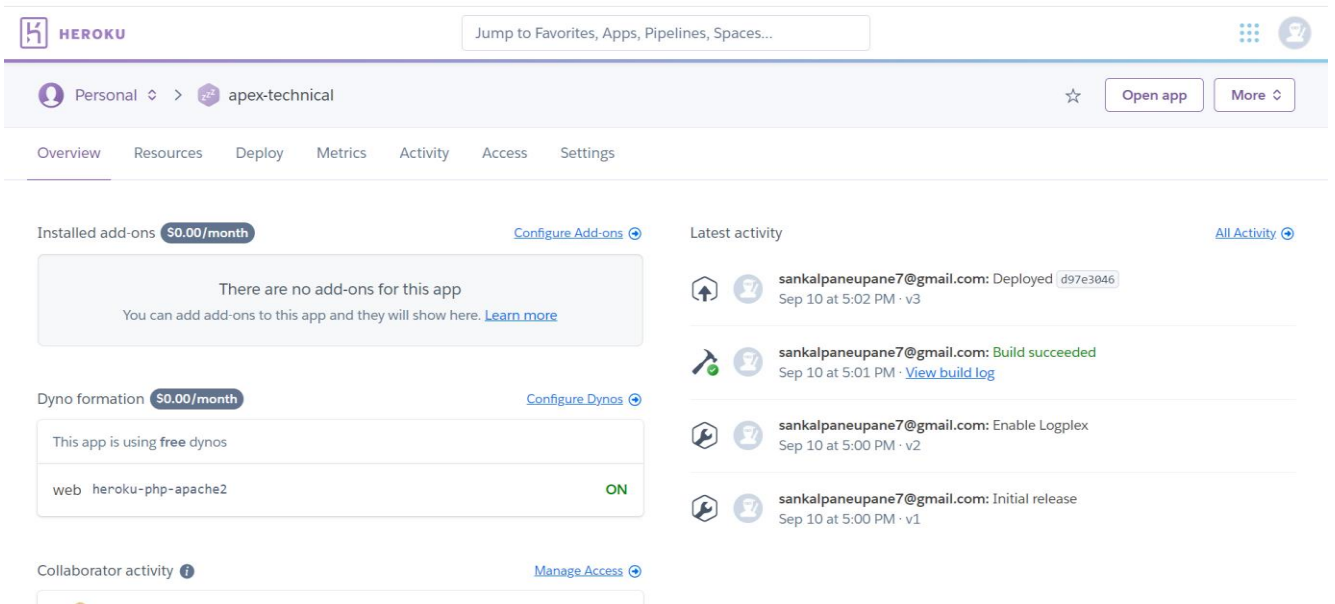


FIGURE 30. Heroku’s Dashboard.

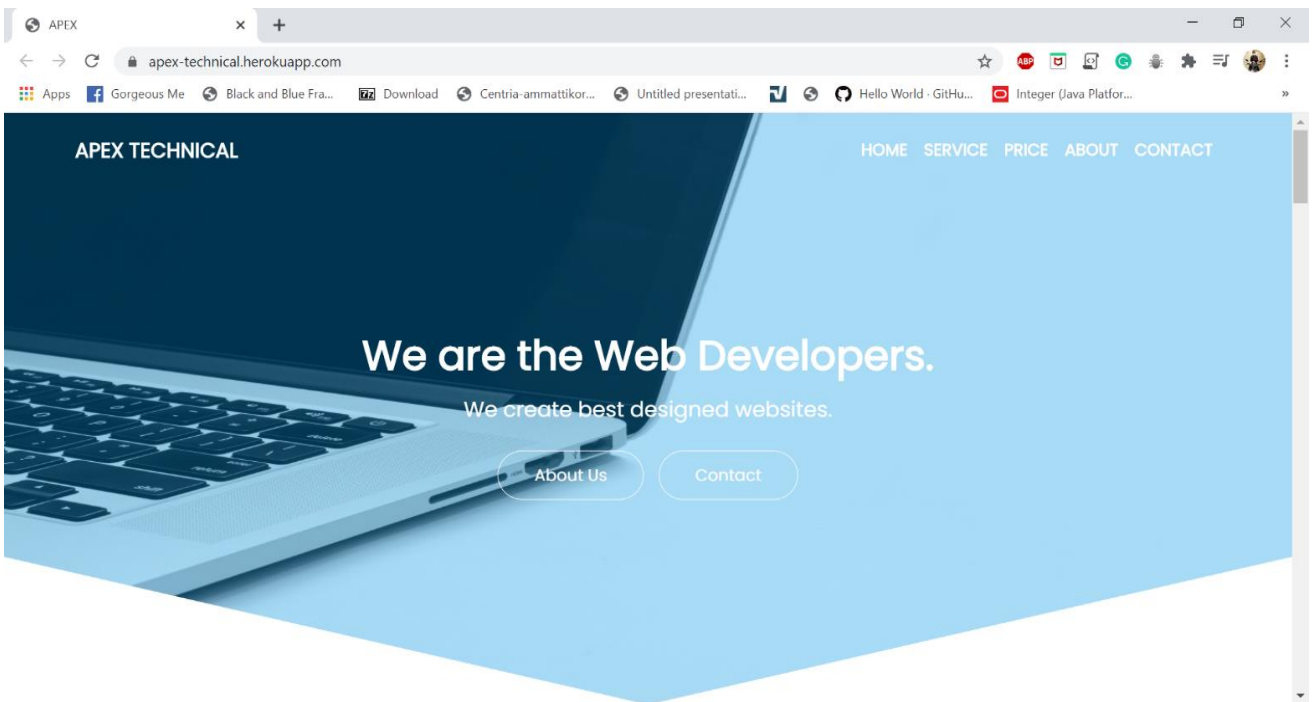


FIGURE 31. Running the web app in the servers using Heroku.

## 6 CONCLUSION

With the advancement of technology and tools, users want to have a website with clear information, fluent and easy navigation. So, the use of the latest technologies to build and deploy the website is the most important part. Also, for this thesis project, the latest technology like HTML 5, CSS 3 and the latest version of JavaScript are used to build the website. On the other hand, the latest version of Heroku is used for the deployment. The reason behind choosing the latest version is to make it available on all sort of browsers without any problem.

The thesis aimed to develop a website and deploy it to a cloud platform, Heroku. This proposal demonstrates in detail the procedure and devices for web application improvement and developments and accordingly can be utilized as an expert guide for comparative purposes. The theory also shows the hypothesis of a cloud platform as help and particularly the accompanying three principal types: PaaS, IaaS, and SaaS. Accordingly, the Heroku cloud stage PaaS was chosen to have an exhibition adaptation of the site, and to fabricate, convey, test, screen, and scale.

Heroku is a cloud platform to build, monitor and scale the app. Heroku has different functionalities, such as accessibility, reliability, and feasibility. Deploying a static website to Heroku was very effective. However, there were a lot of challenges during the process of website deployment such as to use the programming language which is supported by Heroku, using the Git bash terminal, use of commands, changing a static website to a web app. Though it was a challenge throughout the process, Blogs and different YouTube tutorials were beneficial and helpful.

## REFERENCES

- Amazon Web Services. 2020. Cloud Deployment Models. Available: [https://docs.aws.amazon.com/en\\_us/sdk-for-net/v3/ndg/gs-cloud-deployment.html](https://docs.aws.amazon.com/en_us/sdk-for-net/v3/ndg/gs-cloud-deployment.html) . Accessed 14 October 2020
- Andrew, R. 2018. A definition to CSS Framework. Available: <https://www.smashingmagazine.com/2018/11/css-frameworks-css-grid/>. Accessed 13 August 2020
- Ansaharju, J. 2016. Improving software development as P-a-a-S -Using Heroku web application. Helsinki: Metropolia University of Applied Sciences. Master of Engineering thesis. Available: <https://core.ac.uk/download/pdf/38136272.pdf> . Accessed 11 June 2020
- Atos SE. 2020. Deployment to Cloud. Available: [www.apprenda.com/library/cloud/deployment-to-the-cloud/](http://www.apprenda.com/library/cloud/deployment-to-the-cloud/). Accessed 13 October 2020
- Azure.Microsoft.com. 2020. What is a Static Web app? Available:<https://docs.microsoft.com/en-us/azure/static-web-apps/overview>. Accessed 04 October, 2020
- Barton, Z. 2018. What is a cloud application? [Online] Available: <https://www.cloudbakers.com/blog/what-is-a-cloud-application>. Accessed 30 September 2020
- Bootstrap. 2020. Version 4.5 details. Available:<https://blog.getbootstrap.com/2020/05/12/bootstrap-4-5-0/>. Accessed 20 August 2020
- Bootstrap. 2020. How to download? Available: <https://getbootstrap.com/docs/4.5/getting-started/download/>. Accessed 23 August 2020
- Cloudflare Inc. 2020. What is cloud? [Online] Available: <https://www.cloudflare.com/learning/cloud/what-is-the-cloud/>. Accessed 27 September 2020
- Cloudflare Inc. 2020. What is PaaS? Available: <https://www.cloudflare.com/learning/serverless/glossary/platform-as-a-service-paas/>. Accessed 10 October 2020
- Code-Club. 2020. What is javascript? Available: <https://code-mentor.org/javascript-programming-language/>. Accessed 29 August 2020.
- Dummies.com. 2017. Develop a project plan for your website Available: <https://www.dummies.com/web-design-development/develop-a-project-plan-for-your-website/>. Accessed 10 September 2020
- Enterprise Architect Solution. 2019. Application Deployment. Available: [www.enterprise-architecture.org/docs/application\\_architecture/application\\_deployments/](http://www.enterprise-architecture.org/docs/application_architecture/application_deployments/). Accessed 13 October 2020
- Georgiou, M. 2019. Website vs Web apps. Available: <https://www.imaginnovation.net/blog/website-or-web-app-which-is-good-for-business/>. Accessed 02 October 2020
- GeeksforGeeks. 2020. First webpage. Available: <https://www.geeksforgeeks.org/html-course-first-web-page-printing-hello-world/>

- Heroku Inc. 2020. What is Heroku? [online]. San Francisco, CA: Salesforce Inc. Available: <https://www.heroku.com/about>. Accessed 09 June 2020.
- Heroku Inc. 2020. Reference, Heroku Dev Center. [online]. San Francisco, CA: Salesforce Inc. Available: <https://devcenter.heroku.com/categories/reference>. Accessed 09 June 2020
- Heroku Inc. 2020. How Heroku works? [online]. San Francisco, CA: Salesforce Inc. Available: <https://devcenter.heroku.com/articles/how-heroku-works> . Accessed 09 June 2020
- Heroku Inc. 2020. Terminology of Heroku? [online]. San Francisco, CA: Salesforce Inc. Available: <https://devcenter.heroku.com/articles/glossary-of-heroku-terminology> . Accessed 23 July 2019
- Heroku Inc. 2020. Deployment Flow of Heroku. [online]. San Francisco, CA: Salesforce Inc. Available: <https://www.heroku.com/flow> . Accessed 23 June 2019
- Heroku Inc. 2020. Runtime of Heroku. [online]. San Francisco, CA: Salesforce Inc. URL: <https://www.heroku.com/platform/runtime> . Accessed 23 June 2020
- Hissom, E. A. 2011. History and Defination of Html and CSS. Available: <http://amyhissom.com/HTML5-CSS3/history.html#6> . Accessed 12 August 2020
- IBM. 2019. What is PaaS? Available: <https://www.ibm.com/cloud/learn/paas>. Accessed 07 October 2020
- Javascript.info. 2020. Intoduction to Javascript. Available: <https://javascript.info/intro>. Accessed 24 August 2020
- Mont Digital. 2020. What is web application? Available: <https://www.montdigital.com/news/what-is-a-web-application.html>. Accessed 02 October 2020
- Mozilla Developer Network. 2020. Difference between Website and Webpage. Available: [https://developer.mozilla.org/enUS/docs/Learn/Common\\_questions/Pages\\_sites\\_servers\\_and\\_search\\_engines](https://developer.mozilla.org/enUS/docs/Learn/Common_questions/Pages_sites_servers_and_search_engines). Accessed 01 October 2020
- Mozilla Developer Network. 2020. Introduction to JavaScript. Available: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/About\\_JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript). Accessed 29 August 2020
- Mozilla Developer Network. 2020. Definition of CSS. Available: <https://developer.mozilla.org/en-US/docs/Web/CSS>. Accessed 13 August 2020
- Nguyen, L. 2020 Website design and development Turku: Turku university of applied sciences. Bachelor of engineering thesis <https://www.theseus.fi/bitstream/handle/10024/334260/LanNguyenThesis%20.pdf?sequence=2&isAllowed=y> . Accessed 08 August 2020
- Ouellette, A. 2017. What is Bootstrap? Available: <https://careerfoundry.com/en/blog/web-development/what-is-bootstrap-a-beginners-guide/> . Accessed 20 August 2020
- Packt. 2020. Heroku overview. Available: [https://subscription.packtpub.com/book/web\\_development/9781783550975/1/ch011v11sec14/an-overview-of-heroku-s-architecture](https://subscription.packtpub.com/book/web_development/9781783550975/1/ch011v11sec14/an-overview-of-heroku-s-architecture). Accessed 26 June 2020

- Pazos, M. 2018. Wireframing. Available: <https://medium.com/nyc-design/wireframing-f7ce9ee4e703>. Accessed 12 September 2010
- Ranger, S. 2018. What is cloud computing? [Online] Available: <https://www.zdnet.com/article/what-is-cloud-computing-everything-you-need-to-know-about-the-cloud/> . Accessed 27 September 2020
- Red Hat Inc. 2020. What are cloud Services? [Online] Available: <https://www.redhat.com/en/topics/cloud-computing/what-are-cloud-services> . Accessed 30 September 2020
- Rouse, M. 2020. Platform-as-a-Service. Available: <https://searchcloudcomputing.techtarget.com/definition/Platform-as-a-Service-PaaS> . Accessed 10 October 2020
- Rountree, D. & Castrillo, I. 2014. Basic of Cloud Computing. Available: <https://www.sciencedirect.com/topics/computer-science/cloud-deployment-model#:~:text=There%20are%20four%20cloud%20deployment,and%20Infrastructure%20as%20a%20Service>. Accessed 17 October 2020
- Rusev, K. 2018. What is Heroku? Available: <https://mentormate.com/blog/what-is-heroku-used-for-cloud-development/>. Accessed 26 June 2020
- Staticapps.org. 2014. Defining Static Web app. Available: <https://www.staticapps.org/articles/defining-static-web-apps/> . Accessed 05 October 2020
- Violino, B. 2019. What is Platform-as-a-Service? Available: <https://www.infoworld.com/article/3223434/what-is-paas-software-development-in-the-cloud.html> . Accessed 07 October 2020
- Watts, S. & Raza, M. 2019. Difference between IaaS, SaaS and PaaS. Available: <https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/>. Accessed 08 October 2020

