



# **Project Name**

**An Advanced Progressive Web Application - AppBuzz**



## Dedication

I dedicate my project research work to my family, teachers and many friends. A special feeling of gratitude to my loving parents whose words of encouragement and push for tenacity ring in my ears. My parents and teachers always showed me the morals of discipline, honesty and hard-work, helping me deepen the horizon of knowledge and wisdom within me .I also dedicate this dissertation to my many friends who have supported me throughout the process. I will always appreciate all they have done for helping me develop my technology skills.All grandeur and decency to God, for his adequate provision and for engaging the precise individual, by the precise duration through the project.



## ACKNOWLEDGMENT

I would like to express my deepest sincerity and gratitude to the respected (\_\_teacher's Name) for his supervision and guidance that helped me in completing this project successfully. I am forever grateful for their dedication and consistency toward helping us grow.

Similarly, I would like to express gratitude towards my beloved ones, classmates and friends, who always have assisted me through the entire progression.



## **DEFINITION OF TERMS/ ABBREVIATION AND ACRONYMS**

PWA: Progressive Web application

SDK:Software Development Kit

SDLC:Software Development Life Cycle

## Abstract

This coursework is about AppBuzz, a progressive web application (PWA) i.e. a fusion between the features of web pages and mobile apps, specifically built to access variety of features needed in day-to-day lives like weather, news, books, currency(currency rates and converter), quotes and chat (a real-time chat).

AppBuzz is created using node.js with Heroku, a platform as a service (PaaS) that enables developers to build, run, and operate applications entirely in the cloud.

The main objective of creating this Web application is to provide all these features accessible directly from the browser by adding them to Home Screen devices like a typical native app, skipping app marketplaces and saving valuable storage, especially on low-end devices.

AppBuzz has been designed by using Heroku, node.js.

My motivation for this project emerged from the idea to merge features of most used applications into one and to make mobile applications easily accessible and installed in phones.

This project contains all the elaborative details about Progressive Web application module, Features embedded in AppBuzz.

# Table of Contents

<b>Dedication</b>	<b>1</b>
<b>ACKNOWLEDGMENT</b>	<b>2</b>
<b>DEFINITION OF TERMS/ ABBREVIATION AND ACRONYMS</b>	<b>3</b>
<b>Abstract</b>	<b>4</b>
<b>Table of Contents</b>	<b>5</b>
<b>Chapter: 1</b>	<b>6</b>
1.1 Introduction	6
1.2 Existing API Based Applications	8
1.3 Existing PWAs	9
1.4 Declaration of Problem	10
1.5 Projected Solution	10
1.6 Objectives	11
1.7 Questions for Research	11
<b>Chapter: 2</b>	<b>12</b>
2.0 Review of Literature	12
2.1 The concept of PWAs	12
2.2 The concept of API	14
2.3 The concept of WebSocket	15
2.4 The concept of Service Worker	16
2.5 Use of API	17
2.6 Use of WebSockets	17
2.7 Use of Service Worker	18
<b>Chapter: 3</b>	<b>19</b>
3.0 Review of Technology:	19
1.RapidAPI	20
2.New York Times API (NYT API)	21
3.Socket.IO	23
6.PWAs and its components	24
<b>Chapter: 4</b>	<b>26</b>
4.0 Proposed and Applied Methodologies	26
4.1 Product Design	29
4.2 Use case diagram for AppBuzz	29
4.4 Requirement Analysis	30

<b>Chapter: 5</b>	<b>31</b>
5.0 Product Implementation	31
AppBuzz: A Progressive Web Application (Implementation Testing)	31
5.1 Product Testing	35
<b>Chapter: 6</b>	<b>36</b>
6.0 Product Evaluation	36
6.1 Introduction	37
6.2 Potential of the project	38
6.3 Limitation of the product	38
<b>Chapter: 7</b>	<b>39</b>
7.0 Project Management and Evaluation	39
7.2 Value of Project Management	39
7.3 Applied Gantt Chart	40
<b>Chapter 7</b>	<b>40</b>
7.0 Summary	40
7.1 Conclusion	41
<b>Appendices</b>	<b>42</b>
Appendices I: Initial Proposal	42
<b>Maleshkova,Maria.(2010) Investigating web APIs on the World Wide Web:</b>	<b>44</b>
<b>University of Bonn.</b>	<b>44</b>

## Chapter: 1

### 1.1 Introduction

Technology is part of our everyday lives and continuous to revolutionize with time. There is no surprise to see new products launched for different markets in this technological era. Two great companies that started in the 1970s and that have changed the world of computers and technology are Microsoft and Apple.

The web is an incredible platform. Its mix of ubiquity across devices and operating systems, its user-centered security model, and the fact that neither its specification nor its implementation is controlled by a single company makes the web a unique platform to develop software on. Combined with its inherent linkability, it's possible to search it and share what you've found with anyone, anywhere. Whenever you go to a website, it's up-to-date, and your experience with that site can be as ephemeral or as permanent as you'd like. Web applications can reach *anyone, anywhere, on any device* with a single codebase.

**AppBuzz is** a progressive web application (PWA) i.e. a fusion between the features of web pages and mobile apps specifically built to access a variety of features needed in day-to-day lives like weather, news, books, currency(currency rates and converter), quotes and chat ( a real-time chat). It is intended to work on any platform that uses a standards-compliant browser, including both desktop and mobile devices.

AppBuzz consists of WebSocket API, an advanced technology that makes it possible to open a two-way interactive communication session between the user's browser and a server. With this API, you can send messages to a server and receive event-driven responses without having to poll the server for a reply.

AppBuzz also provides key technology powering Progressive Web Application called Service Worker on the mobile web. They allow caching of resources and push notification allowing us to support offline experiences, giving developers complete control over the experience.

To merge features of most used applications into one and to make mobile applications easily accessible and installed in phones, we have included following salient features in this application:

**1.Responsive:** To suit the need for any gadget/hardware-whether it is a desktop, a laptop, phone, or tablet, or any other medium that will be launched in future.

**2.Independent in Connectivity:** PWA (AppBuzz) is supposed to be designed in such a way that it can operate offline or on slow speed networks as well. Therefore, it is not dependent on the nature or speed of the user's connectivity to the internet. AppBuzz is the manifestation of the idea that apps



should be able to run even without an offline server by caching data for the purpose of visual content with the help of Service Worker.

**3.App Based:** Any user can switch to an app-like view or interface on AppBuzz because it offers them an option in the form of “app shell model”. The app shell model enables the design to be split in terms of functionality and content. Thus while the application content will remain the same, it is possible to change or alter the functionalities of the application.

**4.Safety:** Every PWA provides a secure server, the green signal which users can see on top of their browser in the form of a green lock which ensures secure connection

**5.Up-To-Date:** Due to cache-storage ability, AppBuzz is always upgraded timely providing us the features of most used apps like weather, news, books, currency, quotes, chat.

**6.Shareable:** No complicated installation is required for users when they want to operate AppBuzz, all they need to do if they want to share AppBuzz is to share the URL and install it on the phone or use it in browser

**7.Convenient to Keep:** Users can just keep AppBuzz on their home screen and they do not need to access any kind of app store to have access to them.

**8.Display Setting:** AppBuzz has dark mode and light mode as a display setting for user interfaces, such as a smartphone or laptop. It means that, instead of the default dark text showing up against a light screen (known as ‘light mode’), a light colour text (white or grey) is presented against a dark or black screen. Light mode, however, is the default setting for most phones and apps.

## 1.2 Existing API Based Applications

**API** is the acronym for **Application** Programming Interface, which is a software intermediary that allows two **applications** to talk to each other. Each time you use an app like Facebook, send an instant message, or check the weather on your phone, you're using an **API**. Most of the applications today use API.

API's are classified by the systems they are designed for and the availability. So, there are publicly available, partner, and private APIs. For the most part, they are designed for such use cases as:

1. **Database API.** This type helps connect the databases and the database management system.

2. **OS API.** They are intended to provide access to the operating system's resources and functionality. E.g. Cocoa for MacOS.
3. **Remote API.** These provide connectivity between applications on two (or more) different devices. E.g. Java Remote Method Invocation.
4. **Web service API.** Data transfer for web-based apps and services, like PayPal or Google Maps, are the core of the API economy as these APIs are usually shared as public.

Existing Famous Apps that Rely on APIs for Survival:

- **Facebook:** Online social media and social networking service
- **Gmail:** Free email service provided by Google
- **Youtube:** An online video platform owned by Google.
- **Uber:** American technology company for ride-hailing, food delivery, package delivery, couriers
- **Instagram:** An American photo and video sharing social networking service
- **Twitter:** An American microblogging and social networking service
- **Trivago:** Technology company for internet-related services and products in the hotel, lodging and metasearch fields.

Setting a price for API access isn't a new trend. Increasingly, companies have become aware of their software's value, and in some cases they recognize that its integration capabilities are sometimes more important than the service alone. The rise of SaaS (Software as a Service) has made way for the exchange of tools between companies. It's no surprise that the utilities linked to them have evolved into a sort of virtual marketplace. Most of the famous API based Application sets a price for API access because data is the new oil. But, AppBuzz provides all these features for free without any cost.

## 1.3 Existing PWAs

It is a myth that the users will happily download the app of every website they visit frequently. According to Comscore Mobile App Report, over 50% of America's smartphone users download Zero Apps a month. i.e. Each step to download an app reduces 20% of users. PWA reduces the steps between discovery of an app and getting it on the home screen and thereby eliminates friction of getting an app installed. This provides a very fertile ground for businesses to pitch in their PWA.

Some popular progressive Web Applications:

- **Pinterest:**An American image sharing and social media service.
- **Twitter:**An American microblogging and social networking service.
- **Forbes:**American business magazine owned by Integrated Whale Media Investments.
- **Flipkart:**An Indian e-commerce company.
- **Starbuck:**An American multinational chain of coffeehouses and roastery reserves.

We're glued to our phone for an average of 3.15 hours per day, so all these features are necessary to maximize user interaction during that window. But responsive websites can't deliver them. Native apps can, but development costs are higher, and a finished product can be too complex to perform stable enough (thanks, Java). The publisher has to go through an app store to get any traction for the software, and often people are too impatient to spend 10 minutes to install it. That leads to weaker adoption.

A PWA offers enough to substitute for a native app as a budget-friendly option that can be deployed in a matter of days, months, or years, depending on its complexity.

## 1.4 Declaration of Problem

- Native applications are found and installed through an app store, such as Google Play or Apple's iOS App Store which requires separate space for storage.
- Native Apps and specifically for iOS or Android users, this ensures that the experience within the native app is tailor-made to each platform. Developers have to worry about cross-browser or platform compatibility and more on shaping their app for one specific mobile device.
- Most of the native apps require an internet connection to operate. When a native app is installed on a mobile device, it's going to pull directly from the device's resources.
- For "heavier" apps, ones that users interact with frequently, or those they forget to close altogether – resource use in terms of power/battery, storage space and mobile data use can be significant.
- For every single feature like weather, books, currency etc., a separate application is required to download which can be tedious and less efficient.

## 1.5 Projected Solution

- AppBuzz on the other hand, help us avoid dealing with the process of app store submission. Instead, AppBuzz runs on the mobile device's browser. Users access a PWA simply by inputting the URL in the mobile browser and can install it if they want.
- AppBuzz is that the interface of the PWA typically attempts to strike a balance between what you'd find with a responsive website and what you'd encounter in a native app.
- AppBuzz is a web-based app that gets installed on your system and, where possible, works offline utilizing cached data. Service workers are the most important technology allowing offline use in PWAs. Service workers are basically JavaScript files that run independently from the web app itself.
- AppBuzz fixes drainage issues like power/battery, storage space and mobile data.
- One of the most potentially useful things about AppBuzz is that it provides variety of features of different apps in single app For example:
  - ❖ Weather
  - ❖ Chat
  - ❖ Quotes
  - ❖ Books
  - ❖ Currency
  - ❖ News

## 1.6 Objectives

1. Provide a variety of features needed in day-to-day lives like weather, news, books, currency (currency rates and converter), quotes and chat (a real-time chat) all in one app.
2. To build a web application called AppBuzz combining features of both web pages and mobile app.
3. Make features accessible directly from the browser by adding them to Home Screen devices like a typical native app, skipping app marketplaces and saving valuable storage, especially on low-end devices.
4. To create an app that is easily installable in the phone and fixes drainage issues like power/battery, storage space and mobile data.
5. To create Progressive Web Application that can be accessed offline too.
6. To explore features like PWAs, Websockets, APIs, Javascript, Heroku, Service Worker.

## 1.7 Questions for Research

Q1. How does existing literature define Cross-Platform apps, Web apps and Native apps? Q2. How can a Web app be converted into a Progressive Web App?

Q3. How does a Cross-Platform app, Web app and Progressive Web App differ in performance?

Q4. What are the proper lengths for API usages?

Q5. How frequently are APIs used?

## Chapter: 2

### 2.0 Review of Literature

The literature Review describes the search and evaluation of literature in the chosen area such as PWAs, API, Service Worker, WebSockets. It coherently analyses the information and identifies the gaps in current knowledge.

The literature study was conducted in an orderly fashion to ensure a consistent and full depiction of the current state of the literature. The literature involved the examination of journals, conference papers and one blogpost. The blogpost was included due to the fact of it being the first recorded mention of the term Progressive Web App, which is the central concept of this report. The key terms used for searching among the literature included: PWA, Progressive Web App, Progressive Web App Comparison, Cross-platform and Service Worker. The databases used were Researchgate, Google Scholar, Springer, ACM and IEEE. To find additional articles and to make sure that we had covered the most cited research in the area, snowballing methods were also used. In order to know what articles to use from other reference sections, the researchers summaries of previous research were reviewed. This further strengthened the validity in our found research gap, and also ensured that no significant research was overlooked. From the result of the literature review, 16 articles published from 2012 to 2018 were deemed relevant to this report. Due to found inconsistencies regarding certain terms used, we decided to form our own definitions of the terms Native, Cross-platform and Web apps. 7 out of the 16 articles contained different definitions of these terms, described with other sources or with the authors' own definitions. To further strengthen the reality of the issue we also researched how the terms were used in the industry. The motivation behind this was only to make sure that the same problem was present in the industry, we settled with three different blog posts. Even here, the same inconsistencies were apparent. This research therefore felt the need to define our own definitions of the terms for two reasons. Firstly, to avoid confusion for the reader when presented with the terms. This also gave us as writers freedom to use the terms as we pleased throughout the text, knowing that the reader will not feel any contradiction regarding the perception of them. Secondly, we wanted to contribute and show regard to the researchers before us who had spent time and effort to create

their definitions of the terms. Our definitions are based upon the 5 out of 7 articles we found defining their own terms.

## 2.1 The concept of PWAs

### Progressive Web Application

Application software (app for short) is computing software designed to carry out a specific task other than one relating to the operation of the computer itself, typically to be used by end-users. A progressive web Application (PWA) is a type of application software delivered through the web, built using common web technologies including HTML, CSS and JavaScript. It is intended to work on any platform that uses a standards-compliant browser, including both desktop and mobile devices.

Since a progressive web app is a type of webpage or website known as a web application, they do not require separate bundling or distribution. Developers can just publish the web application online, ensure that it meets baseline "installability requirements", and users will be able to add the application to their home screen. Publishing the app to digital distribution systems like Apple App Store or Google Play is optional.

It is a myth that the users will happily download the app of every website they visit frequently. According to Comscore Mobile App Report, over 50% of America's smartphone users download Zero Apps a month. i.e. Each step to download an app reduces 20% of users. PWA reduces the steps between discovery of an app and getting it on the home screen and thereby eliminates friction of getting an app installed. This provides a very fertile ground for businesses to pitch in their PWA.

According to the study, mobile web reach is way higher than native app reach. It was 11.4 million unique visitors per month compared to 4 million visitors. Whereas the stats of user engagement with services showed that users tend to spend more time on native mobile apps compared to the standard web app. It was an average of 188.6 minutes on app against 9.3 minutes on the web. So, the idea was clear. They wanted to provide a native app like engaging experience to users on the mobile web. In this way, Progressive Web Apps were developed to deliver amazing user experience on the web.

Russell and Berriman defined the following attributes for PWAs in 2015 : -

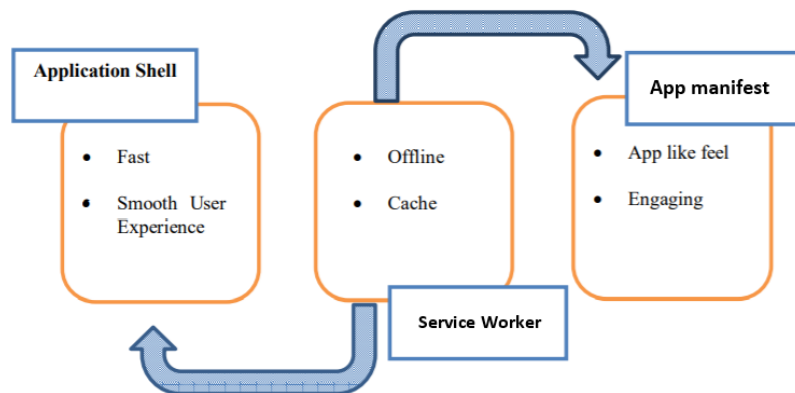
- **Responsive:** Should be built in order to work smoothly on a wide variety of devices
- **Connectivity independent:** Should take advantage of Service Workers in order to function even without a network connection
- **App-like interactions:** Should use an application shell architecture and dynamically load content
- **Fresh:** Content should be up to date
- **Safe:** Should use Transport Layer Security
- **Discoverable:** System should recognize them as applications
- **Re-engageable:** Should use system APIs such as push notifications to re engage the user

- **Installable:** After a certain amount of interaction with the user, the app can be added to the home screen - **Linkable:** Should be easy to install and share.

Below is the comparison between Native App, PWA and Standard web app on various important parameters:

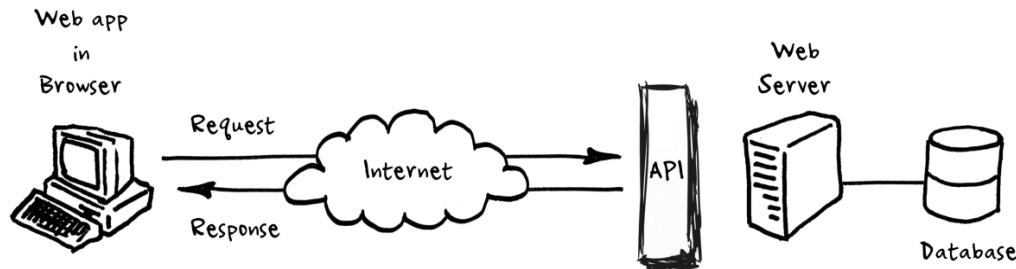
	Native App	PWA	Standard web App
<b>Installation</b>	Need to go to the App store or Play Store, click download	Just click a button to add them to their phone home screen (only on Android)	Installation not required
<b>Updates</b>	Need to be submitted to the store, then downloaded by the user	Updates are instant	Updates are instant
<b>Size</b>	Mostly heavy in size. They can take time for downloading on a users' device	Small and fast	Small and fast
<b>Offline access</b>	Available	Need to use the app once online, then should be able to access the cached content offline	Not required
<b>User experience</b>	Excellent when the application is well designed	Confusing because of the double menus (app menu and browser menu)	Same as progressive web app
<b>Push notification</b>	Yes	Yes (Android Only)	Yes (Only possible with third party services)
<b>Discoverability</b>	Not good-need to work hard on app store optimization	Good – to make appear in search results, need to be optimized for SEO	Not required

### Core Tenets of PWA



## 2.2 The concept of API

### Application Programming Interface(API)



An application programming interface (API) is an interface that defines interactions between multiple software applications or mixed hardware-software intermediaries. It defines the kinds of calls or requests that can be made, how to make them, the data formats that should be used, the conventions to follow, etc. It can also provide extension mechanisms so that users can extend existing functionality in various ways and to varying degrees. An API can be entirely custom, specific to a component, or designed based on an industry-standard to ensure interoperability. Through information hiding, APIs enable modular programming, allowing users to use the interface independently of the implementation.

API lets a developer make a specific “call” or “request” in order to send or receive information. This communication is done using a programming language called “JSON.” It can also be used to make a defined action such as updating or deleting data.

### Why API?

In building applications, an API (application programming interface) simplifies programming by abstracting the underlying implementation and only exposing objects or actions the developer needs. While a graphical interface for an email client might provide a user with a button that performs all the steps for fetching and highlighting new emails, an API for file input/output might give the developer a function that copies a file from one location to another without requiring that the developer understand the file system operations occurring behind the scenes.

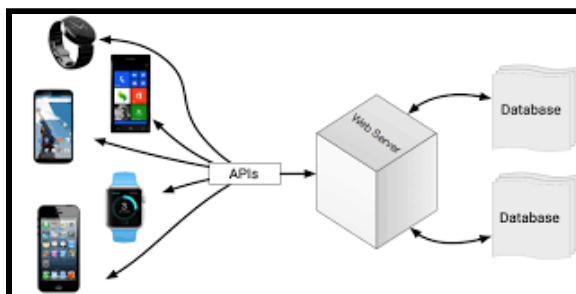


Fig: API programming: Backbone of Mobile App Development

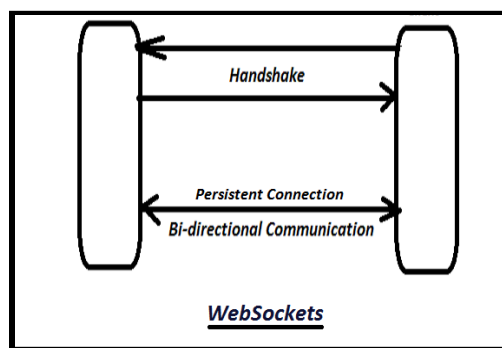


## 2.3 The concept of WebSocket

### WebSocket

A WebSocket is a persistent connection between a client and server. WebSockets provide a bidirectional, full-duplex communications channel that operates over HTTP through a single TCP/IP socket connection. At its core, the WebSocket protocol facilitates message passing between a client and server. This article provides an introduction to the WebSocket protocol, including what problem WebSockets solve, and an overview of how WebSockets are described at the protocol level.

The WebSocket protocol enables interaction between a web browser (or other client application), facilitating real-time data transfer from and to the server. This is made possible by providing a standardized way for the server to send content to the client without being first requested by the client, and allowing messages to be passed back and forth while keeping the connection open.



### Why WebSocket?

The idea of WebSockets was borne out of the limitations of HTTP-based technology. With HTTP, a client requests a resource, and the server responds with the requested data. HTTP is a strictly unidirectional protocol — any data sent from the server to the client must be first requested by the client. Long-polling has traditionally acted as a workaround for this limitation. With long-polling, a client makes an HTTP request with a long timeout period, and the server uses that long timeout to push data to the client. Long-polling works, but comes with a drawback — resources on the server are tied up throughout the length of the long-poll, even when no data is available to send.

WebSockets, on the other hand, allow for sending message-based data, similar to UDP, but with the reliability of TCP. WebSocket uses HTTP as the initial transport mechanism, but keeps the TCP connection alive after the HTTP response is received so that it can be used for sending messages between client and server. WebSockets allow us to build “real-time” applications without the use of long-polling.

## 2.4 The concept of Service Worker

**SERVICE WORKERS:** Service Workers, an incredibly powerful tool behind Progressive Web App.

A **service worker** is a script that your browser runs in the background, separate from a web page, opening the door to features that don't need a web page or user interaction.

The features provided by service workers are:

- Offline Access.
- Push Notifications
- Background content updating

Content caching Service workers perform the following functionalities:

1. Caches the App Shell.
2. Updates the Content in background.
3. Gets the push notification id from the user to send the notification.
4. Invalidates the cache when needed

**APP SHELL:** Application Shell Architecture is served up by the Service Worker and then the content is delivered. These are often cached by the service worker from its source through API requests. The sites that people visit more often will be able to hold the last content the person visited while waiting for the network to dynamically load the latest refresh. With the App Shell model, the focus is on keeping the shell of app UI and the content inside of it separate, and they are cached separately. Ideally, App Shell is cached such that it loads as quickly as possible when a user visits and returns later. Having the shell and the content load separately theoretically improves the user's perception of the performance and usability of the app.

## 2.5 Use of API

In building applications, an API (application programming interface) simplifies programming by abstracting the underlying implementation and only exposing objects or actions the developer needs. While a graphical interface for an email client might provide a user with a button that performs all the steps for fetching and highlighting new emails, an API for file input/output might give the developer a function that copies a file from one location to another without requiring that the developer understand the file system operations occurring behind the scenes.

## 2.6 Use of WebSockets

The WebSocket API is an advanced technology that makes it possible to open a two-way interactive communication session between the user's browser and a server. With this API, you can send messages to a server and receive event-driven responses without having to poll the server for a reply.

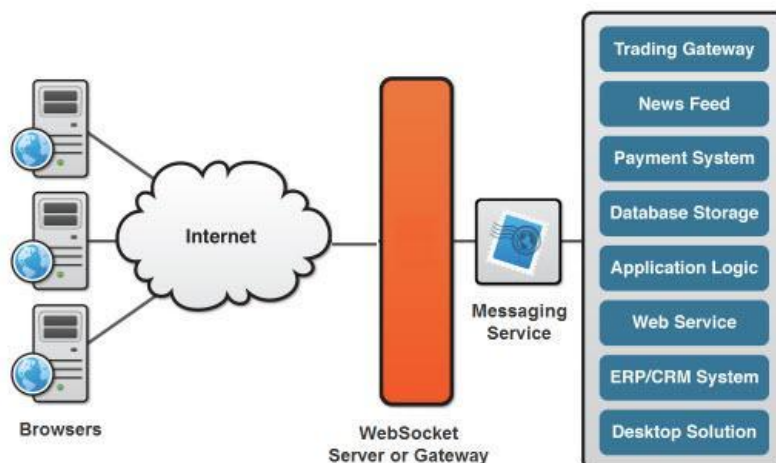
The WebSocket protocol enables interaction between a web browser (or other client application), facilitating real-time data transfer from and to the server. This is made possible by providing a standardized way for the server to send content to the client without being first requested by the client, and allowing messages to be passed back and forth while keeping the connection open.

### Interfaces

**WebSocket:** The primary interface for connecting to a WebSocket server and then sending and receiving data on the connection.

**CloseEvent:** The event sent by the WebSocket object when the connection closes.

**MessageEvent:** The event sent by the WebSocket object when a message is received from the server.



## 2.7 Use of Service Worker

Service workers enable applications to control network requests, cache those requests to improve performance, and provide offline access to cached content.

Service workers depend on two APIs to make an app work offline: Fetch (a standard way to retrieve content from the network) and Cache (a persistent content storage for application data). This cache is persistent and independent from the browser cache or network status.

Act as the base for advanced features

- Service workers provide the starting point for features that make web applications work like native apps. Some of these features are:
- Notifications API: A way to display and interact with notifications using the operating system's native notification system.
- Push API: An API that enables your app to subscribe to a push service and receive push messages. Push messages are delivered to a service worker, which can use the information in the message to update the local state or display a notification to the user. Because service workers run independently of the main app, they can receive and display notifications even when the browser is not running.
- Background Sync API: Lets you defer actions until the user has stable connectivity. This is useful to ensure that whatever the user wants to send is actually sent. This API also allows servers to push periodic updates to the app so the app can update when it's next online
- Channel Messaging API: Lets web workers and service workers communicate with each other and with the host application. Examples of this API include new content notification and updates that require user interaction.

## Chapter: 3

### 3.0 Review of Technology:

In the technical environment, we may encounter numerous tools for developing a product. Some may contain the same properties while other tools and some may be having unique functionality of a tool. Identifying the functionality of a tool is a critical aspect when developing a product because these tools may have similar properties but may have different features within that makes them unique in their own way. The required features for developing our product might be scattered within different tools, so we must identify and allocate the only tools required for the developing of our product. In this section, we will be identifying and distinguishing various tools that were adopted to make this project.

Brief introduction of the tools that were required for developing this Progressive Web Application: AppBuzz are illustrated below:

#### 1.RapidAPI

RapidAPI is an API Marketplace for developers to find, connect, and manage their API connections. Find the APIs that you need for your project, embed the API into your app, and track usage of all your APIs through a single dashboard. If you have an API you've created, use RapidAPI to make it available to hundreds of thousands of developers already utilizing APIs through RapidAPI.

#### For App Developers

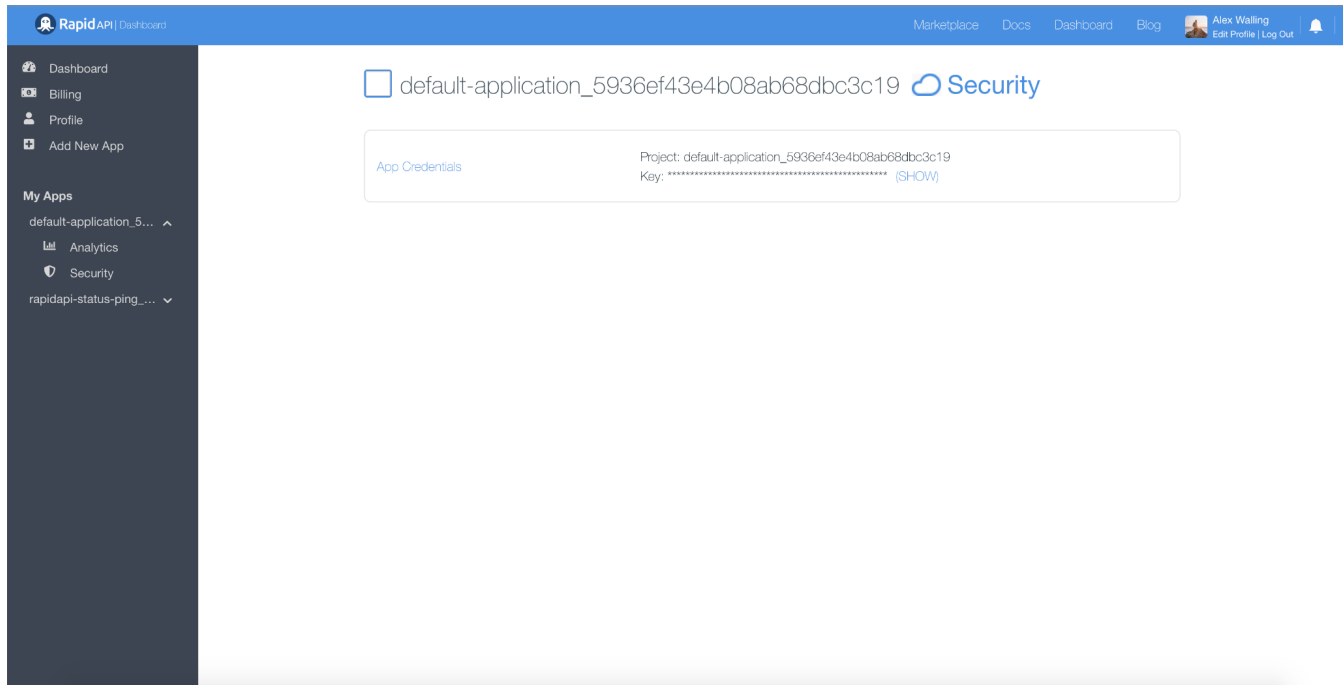
It's really easy to use RapidAPI to start consuming APIs in your app. Using RapidAPI, you can consume any API using a unified, REST format that is easy to understand and embed in your app. Moreover, you can view all of the APIs you are connected to using the dashboard, which monitors things like the number of API requests, latency, and error rates.

#### API Keys

When connecting an API to a project or application, you must have an API key to authenticate your request. Creating an app within RapidAPI generates an API key (X-RapidAPI-Key) specific to that application. You can view analytics based on the API calls you make using this app key.

## Generating a New App and Finding API Keys

To generate a new app, which also generates new API keys, or locate the API keys for one of your apps, head over to your RapidApi developer dashboard.



From here you can either 'Add New App' or select the 'Security' tab of an application to find your API keys.

## 2.New York Times API (NYT API)

Since, we have included features in Appbuzz such as News and Books for up-to-date information among the users. We have chosen the famous New York Times as our news source. We used the New York Times API because NYT is one of reliable news sources and also because the documentation was quite detailed. Furthermore, we use API from New York times to show Books in category “Best Seller of the Week” which can be purchased via various sites.

## Using the NYTimes API

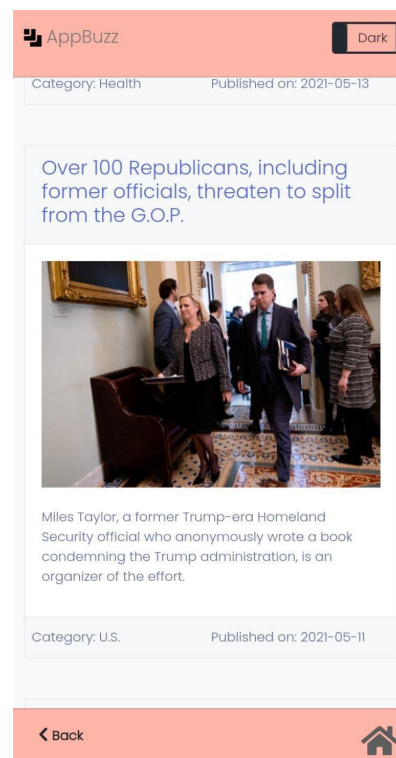
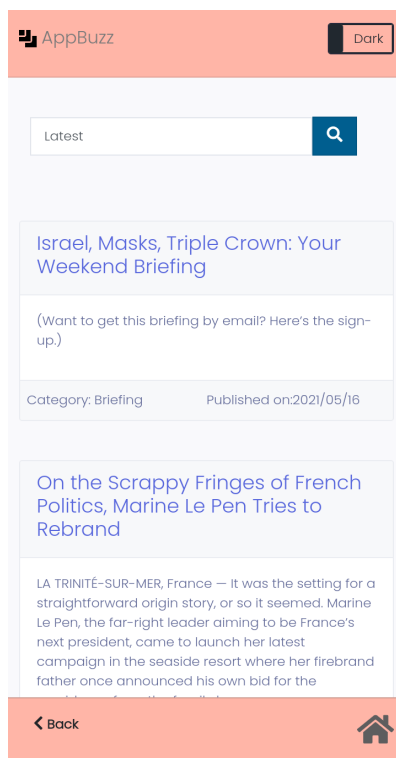
First, create a developer account so that you can get an api key for your project.

- Create your developer account at <https://developer.nytimes.com/>
- Create an app to get an api key at <https://developer.nytimes.com/my-apps>

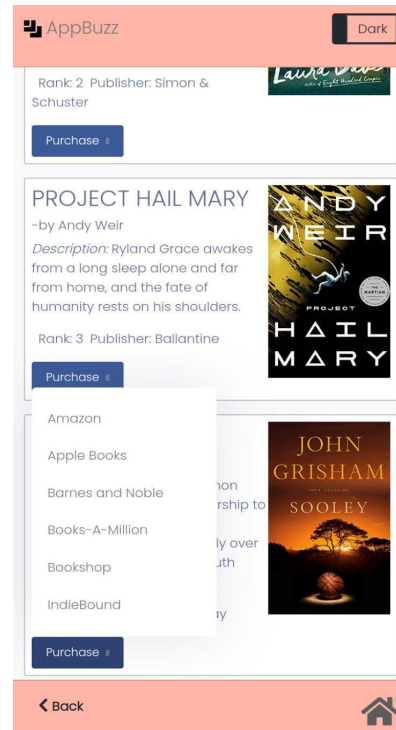
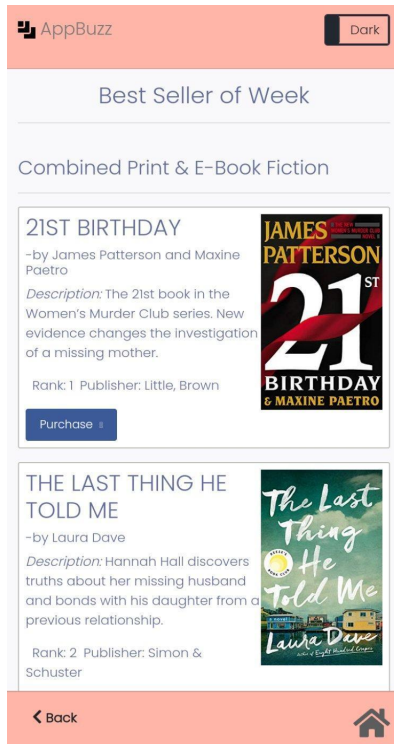
The developer page provides 12 APIs including Archive, Article Search, Geo, Top Stories, Books and Movie Reviews APIs. Among them we have used Article(news), Books,Search in AppBuzz.

App looks like this for NEWS;

:



App looks like this for BOOKS:



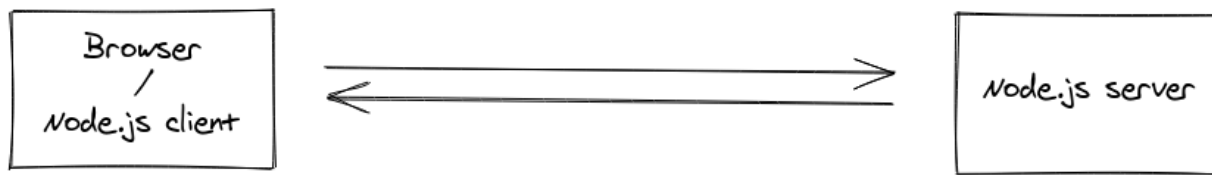
### 3.Socket.IO

#### What is Socket.IO ?

Socket.IO is a library that enables real-time, bidirectional and event-based communication between the browser and the server. It consists of:

- a Node.js server: [Source](#) | [API](#)
- a Javascript client library for the browser (which can be also run from Node.js): [Source](#) | [API](#)





### How does that work?

The client will try to establish a WebSocket connection if possible, and will fall back on HTTP long polling if not.

WebSocket is a communication protocol which provides a full-duplex and low-latency channel between the server and the browser.

- The browser supports WebSocket (97% of all browsers in 2020)
- There is no element (proxy, firewall, ...) preventing WebSocket connections between the client and the server

### Why Socket.IO?

Here are the features provided by Socket.IO over plain WebSockets:

- reliability (fallback to HTTP long-polling in case the WebSocket connection cannot be established)
- automatic reconnection
- packet buffering
- acknowledgments
- broadcasting to all clients or to a subset of clients (what we call "Room")
- multiplexing (what we call "Namespace")

## 4. WebBased Geolocation

The WebBased Geolocation API allows the user to provide their location to web applications if they so desire. For privacy reasons, the user is asked for permission to report location information.

WebExtensions that wish to use the Geolocation object must add the "geolocation" permission to their manifest. The user's operating system will prompt the user to allow location access the first time it is requested.

You will often want to retrieve a user's location information in your web app, for example to plot their location on a map, or display personalized information relevant to their location.

The Geolocation API is accessed via a call to `navigator.geolocation`; this will cause the user's browser to ask them for permission to access their location data. If they accept, then the browser will use the best available functionality on the device to access this information (for example, GPS).

**Feature:**

Geolocation allows you to not only keep track of up-to-date geographical information but also perform turn-by-turn navigation to guide users to a specific location or display points of interest that are within their reach. This function can be used to indicate a route. It is useful for geotagging custom content.

## 6.PWAs and its components

### WHAT IS A PWA?

“Progressive Web Apps (PWA) are built and enhanced with modern APIs to deliver enhanced capabilities, reliability, and installability while reaching anyone, anywhere, on any device with a single codebase.”

— Google’s developers

Russell and Berriman defined the following attributes for PWAs in 2015 : -


- **Responsive:** Should be built in order to work smoothly on a wide variety of devices
- **Connectivity independent:** Should take advantage of Service Workers in order to function even without a network connection
- **App-like interactions:** Should use an application shell architecture and dynamically load content
- **Fresh:** Content should be up to date
- **Safe:** Should use Transport Layer Security
- **Discoverable:** System should recognize them as applications
- **Re-engageable:** Should use system APIs such as push notifications to re engage the user
- **Installable:** After a certain amount of interaction with the user, the app can be added to the home screen - Linkable: Should be easy to install and share.

In other words, PWAs are websites that users can use as stand-alone applications. They are different from native apps mainly because PWAs don’t require installation and can be used with various devices — native apps are primarily built for mobile devices.

### HOW DO PWA WORK?

The core of a PWA consists of three components:

a web app manifest, service workers, and an application shell.



Here's what these components do.

### **Web App Manifest**

The web app manifest is the core for making a website run as a stand-alone application in full-screen mode. You can define how the PWA looks, optimize it for different devices, and assign an icon that's displayed after the application's installation.

### **Service Workers**

The service workers enable the offline usage of the PWA by fetching cached data or informing the user about the absence of an Internet connection. The service workers also retrieve the latest data once the server connection is restored.

### **Application shell architecture**

The application shell is what the users see when they access a PWA. It's the minimum HTML, CSS, and JavaScript that's required to power the user interface. When developing a PWA, you can cache the application shell's resources and assets in the browser.

## **Chapter: 4**

### **4.0 Proposed and Applied Methodologies**

The format and the nature of the project defines the use of certain methodologies. A methodology is a practice or moreover a technique with procedure and rules that binds the work in discipline. Choosing a correct methodology is very important while creating a product because the development of product bended by the rules results in a quality of product. A methodology that is being rooted in some vital works defines why we choose to do things in a certain way.

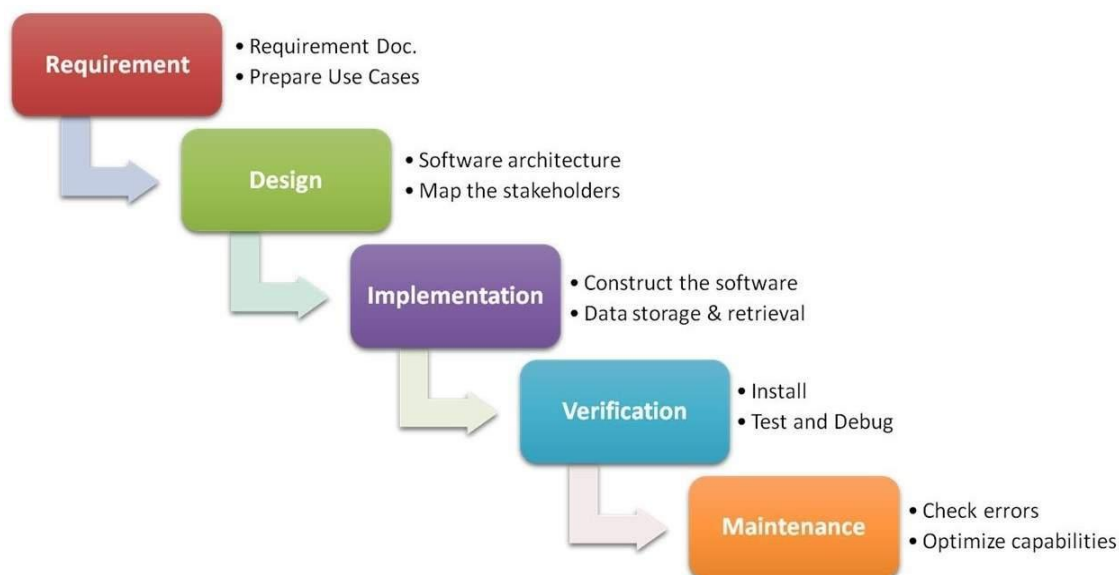
A project can be delivered in many different ways. And using a correct methodology within means making a structural plan for the process of developing a product. There are different methodology frameworks that could be applied to a project i.e Agile method and Waterfall method and in this project, I have researched to apply the Waterfall method.

Agile Methodology refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing

cross-functional teams. Agile methods or Agile processes generally promote a disciplined project management process that encourages frequent inspection and adaptation, a leadership philosophy that encourages teamwork, self-organization and accountability, a set of engineering best practices intended to allow for rapid delivery of high-quality software, and a business approach that aligns development with customer needs and company goals.

The waterfall methodology is a linear project management approach, where stakeholder and customer requirements are gathered at the beginning of the project, and then a sequential project plan is created to accommodate those requirements. The waterfall model is so named because each phase of the project cascades into the next, following steadily down like a waterfall.

Whatever the scenario is, both waterfall and agile methods are the system development lifecycle methodologies and are abided by the rules of Software Development Life Cycle (SDLC) phases: planning, Analysis, Design, Implementation and Maintenance. A waterfall model is one of the popular methods of SDLC. Often known as a traditional approach to the system development life cycle, it describes the development of the product to be rigid and more linear. There is no turning back in this methodology, as the product is developed on certain goals and phases where on completion of each phase the next ones get started. It means that the outcome from one of the phases is the input for the next phase.



**Fig: Waterfall Methodology in Software Development**

The sequential phases in Waterfall model are –

- Requirement Gathering and analysis – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- System Design – The requirement specifications from the first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- Implementation – With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- Verification – All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures. Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- Maintenance – There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for the previous phase and it is signed off, so the name "Waterfall Model". In this model, phases do not overlap.

As this is a solo project, I will use a waterfall model. In this model, one phase must be completed to enter another phase. It means if I completed requirement gathering then only I will work on designing which is the beauty of the waterfall model. In the agile model, all the phases start at the same time, which is teamwork, but in this project, there are no any team members. Therefore, I go through the waterfall model.

Waterfall Model is chosen over Agile methodology for reasons mentioned below:

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.
- Project is done by solo member

## 4.1 Product Design

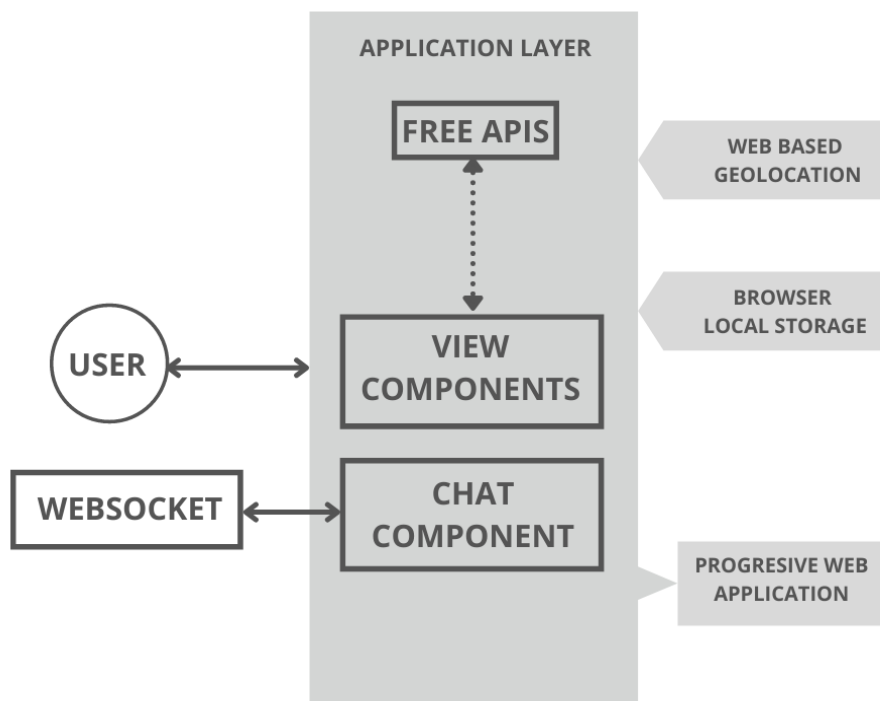
In the Design phase of the System Development Life-Cycle, the System is designed in an appropriate format referencing the requirements that have been listed in the Planning phase of SDLC. The product designing will be the crucial part in this phase. We take note of the looks of the project and mockup the overall project. Listing the functionality of various sections of product, specifying the rules and designing the section will be the major task in this phase. Additionally, the product design phase carries out the task of consolidating different modules and adding functional specification to run the project.

The design part is classified into two groups, i.e.: system design and component design. The overall detailed information of the system explaining how well the components interact with each other is done in the system design phase of SDLC. The information and feedbacks generated by the clients are collected in this design phase along with the fundamental design such as: devices used, security check, data type, flow of data over system for defining certain features and operations like user information, rules, diagrams and procedures and functions. The component design in the SDLC shows the functioning of the system and components and their results generated after full completion of the products. The

process of input, output, databases and files, and main components are further looked at and analyzed and designed with proper diagrams like Sequence diagram, component diagram and data flow diagram.

## 4.2 Use case diagram for AppBuzz

A **use case diagram** is a graphical depiction of a user's possible interactions with a system. A **use case diagram** shows various **use cases** and different types of users the system has and will often be accompanied by other types of **diagrams** as well.



4.4

## Requirement Analysis

Requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements. In other words, Requirement analysis is the process of describing the user's expectation from the new system. It is also known as information gathering or requirement capturing. It circumscribes those works that need to go into determining the needs and conditions for meeting the project.

This phase aims to deal with specified documents so that the requirements are fair and structured. Deciding the requirement that will be involved is important, as it involves the use of traditional methods of information collection, analyzing the documents and other parts of the system which would be simple, reliable and profitable. The work is done when there is very less risk involved in the project. Developing the system involved analyzing the system requirements. As the system grows with

additional features and requirements it gets complex and the goal would be easily understood. Thus, the process of requirement analysis begins.

### **1) FUNCTIONAL REQUIREMENTS**

In software engineering, a functional requirement defines a system or its component. It describes the functions a software must perform. A function is nothing but inputs, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform.

In the case of this system of AppBuzz, it is developed so that the users are able to attain up-to-date information all in one app and can interact with others creating chat rooms. The users have their own username and room name which helps them to create room for communication. Admin controls the overall system. The admin will be able to create, update or delete any users, events, images, stories, news, chatrooms, quotation and other features if needed. Admin gets the privilege of controlling the entire application. Users will be able to check news, buy books, check currency and convert it, check useful quotations, chat with their friends. By accessing device location, users will be able to check weather as well. These features can be accessed offline too once cache is fetched. Dark mode and normal mode are available for the convenience of the user.

### **2) Non-Functional Requirements**

A non-functional requirement defines the quality attribute of a software system. They represent a set of standards used to judge the specific operation of a system. Example, how fast does the website load?

A non-functional requirement is essential to ensure the usability and effectiveness of the entire software system. Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. This is the requirement that is as useful as functional requirements in the system because it specifies the overall standards used for reviewing the system. It is also defined as system qualities requirements. They define the usability and effectiveness of the entire system. If the system fails to achieve any one of the non-functional requirements, they result in system failure to meet the users, business or market needs. The Non-functional requirements have the type of managing requirement like: functionality, usability, reliability and performance (Eriksson, U 2012).



## Chapter: 5

### 5.0 Product Implementation

During the implementation phase, the project takes the shape. The conversion of actual requirements and objectives into actual product is done in this phase. The programmer are engaged in coding, designing and re-testing the project. It is the phase when the project becomes visible to everyone

#### AppBuzz: A Progressive Web Application (Implementation Testing)



**Figure 1: Overview**

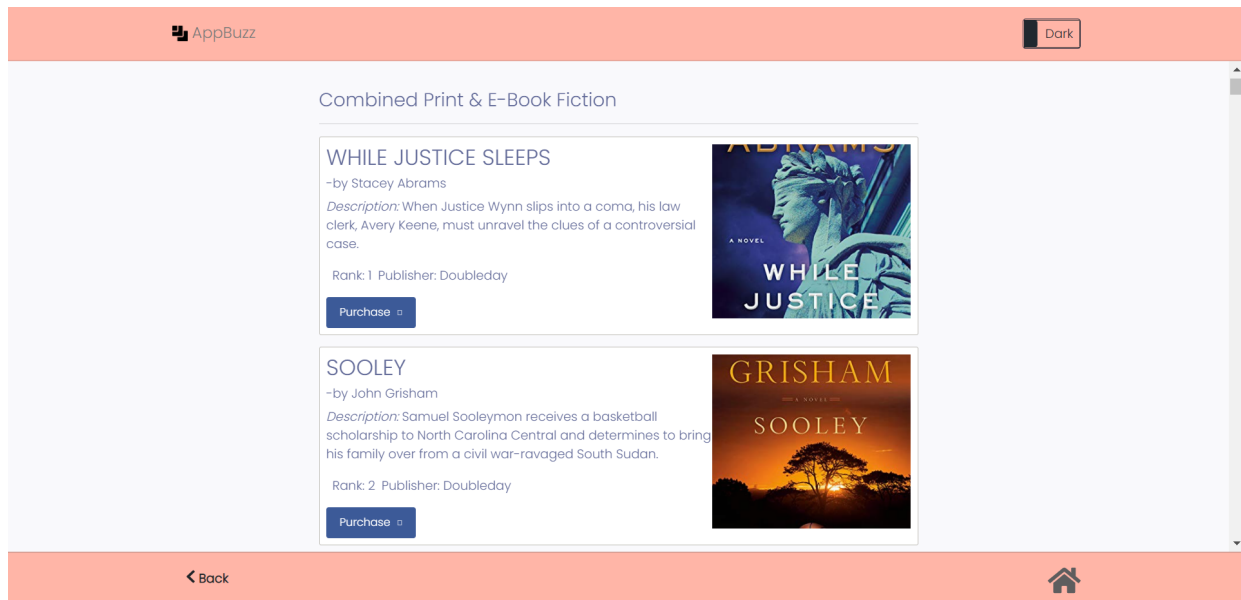


Figure 2: Books Section

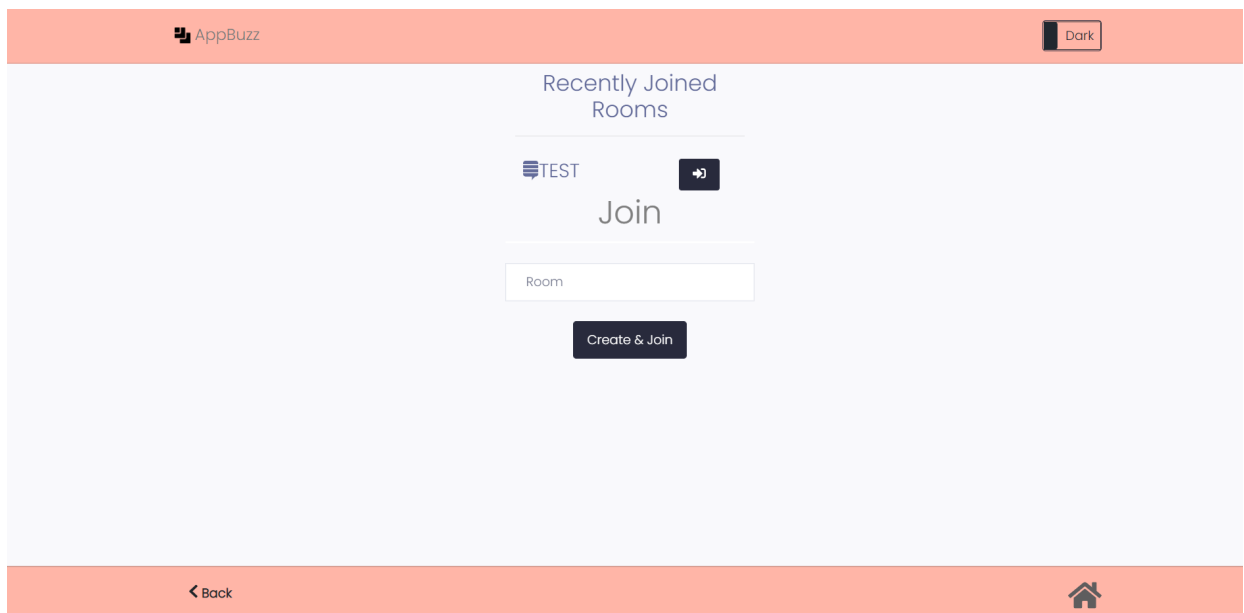


Figure 3: Chat Section

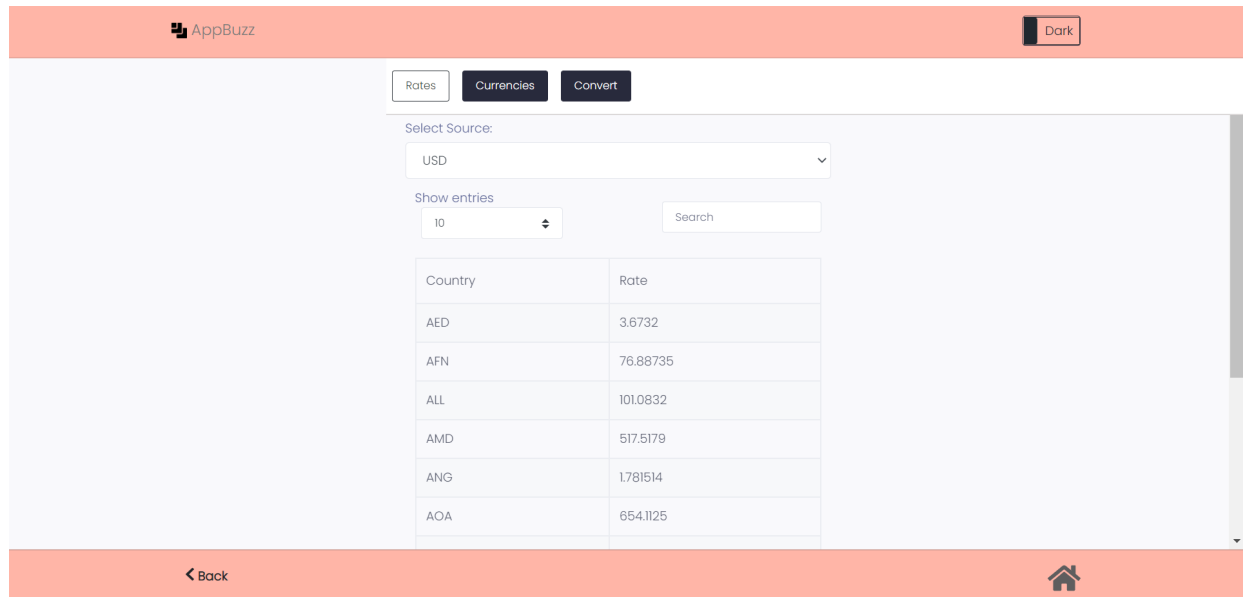


Figure 4: Currency Section

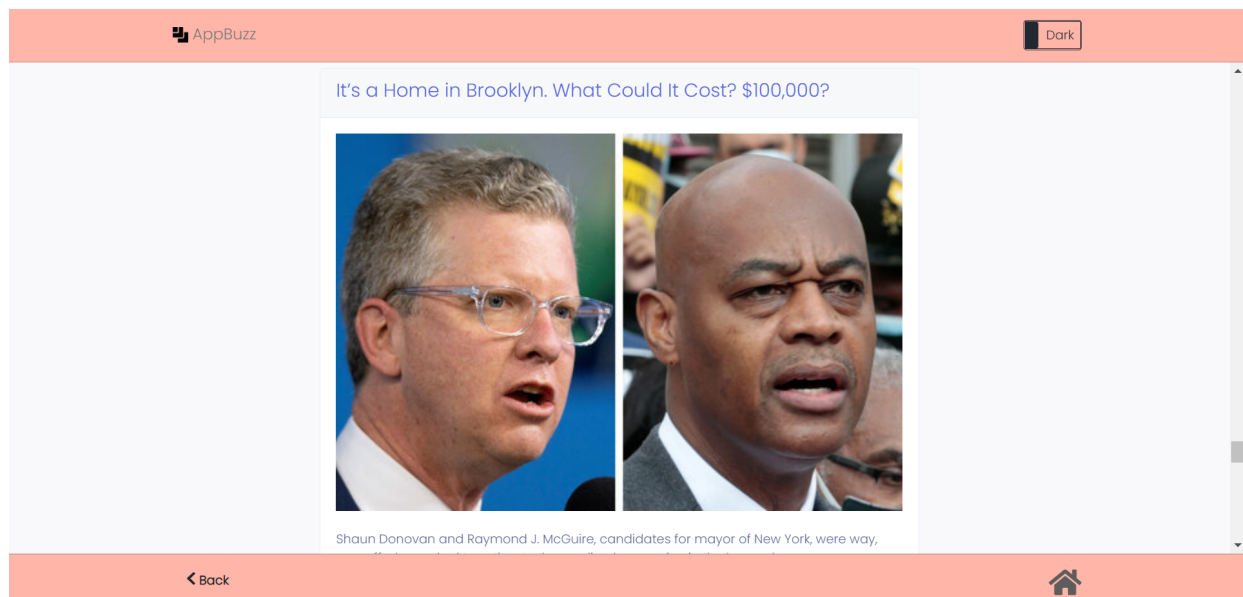


Figure 5: NEWS Section

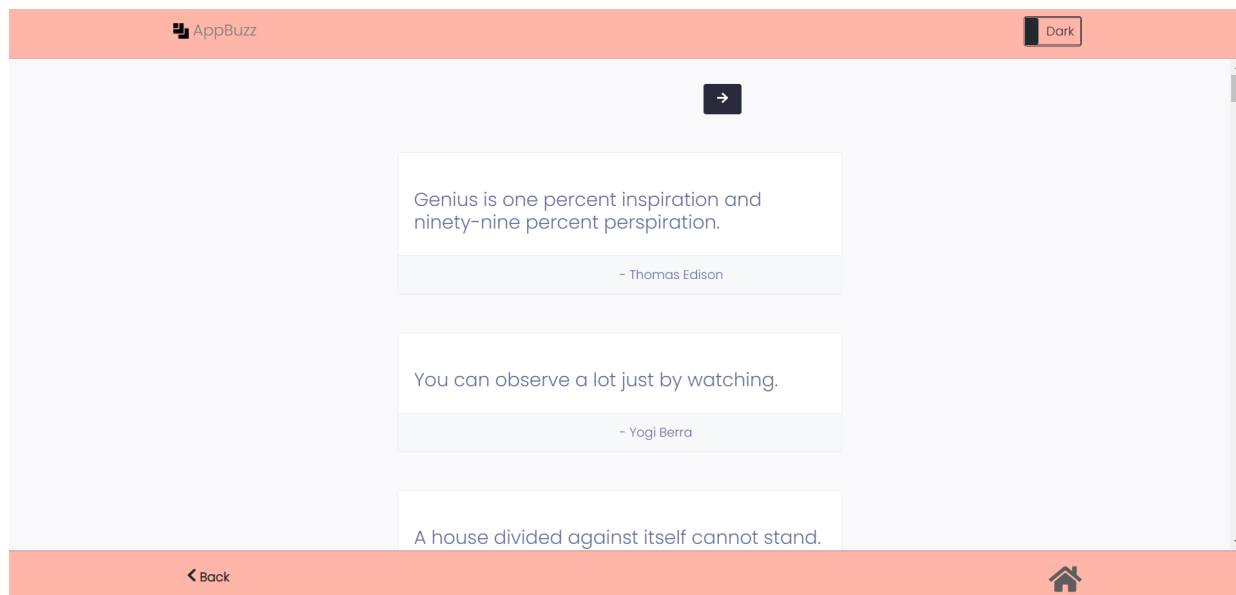


Figure 6: Quotes Section

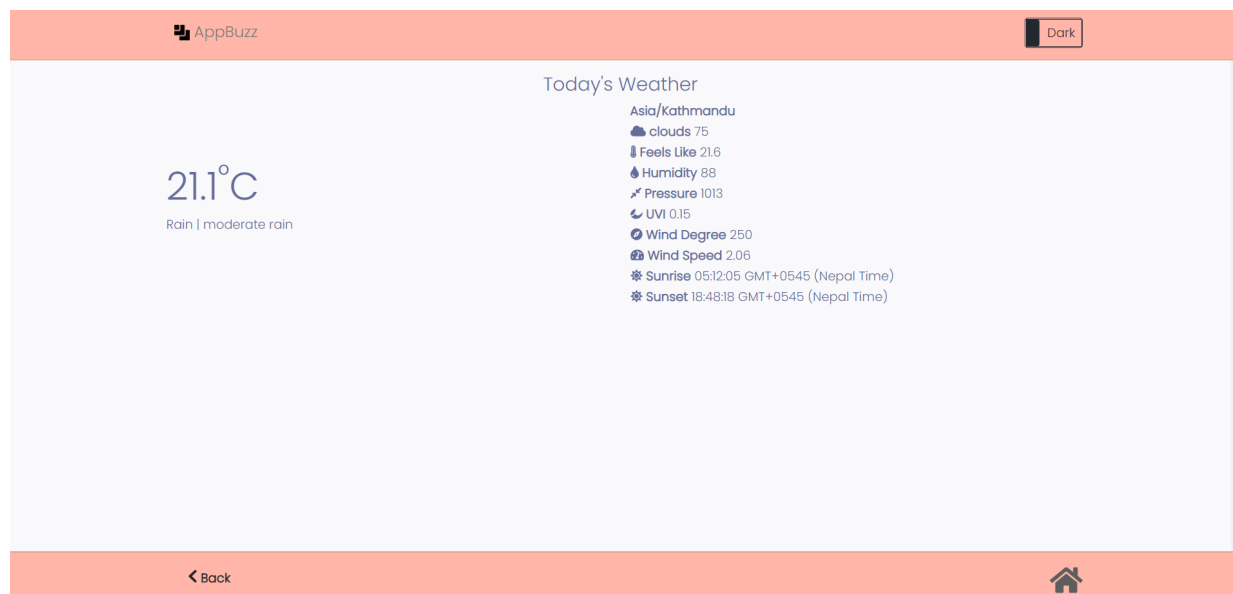


Figure 7: Weather Section

## 5.1 Product Testing

Here, it is tested how APIs are used in application and chat components.

The first screenshot shows the AppBuzz application interface with a 'Rates' tab selected. The 'Select Source' dropdown is set to 'USD'. A table displays exchange rates for various countries:

Country	Rate
AED	3.6732
AFN	76.88735
ALL	101.0832
AMD	517.5179
ANG	178.034
AOA	654.1125
ARS	93.08828
AUD	1.27486
AWG	1.80025

The second screenshot shows the browser's developer tools with the Network tab open. The selected request is 'overview.json?api-key=o20RJ...'. The response is a JSON object containing exchange rates for various currencies.

```
{
  "timestamp": 1628549384852,
  "source": "USD",
  "date": "2021-05-9",
  "rates": {
    "AED": 3.6732,
    "AFN": 76.88735,
    "ALL": 101.0832,
    "AMD": 517.5179,
    "ANG": 1.781514,
    "AOA": 654.1125,
    "ARS": 93.08828,
    "AUD": 1.27486,
    "AWG": 1.80025,
    "AZN": 1.7,
    "BAM": 1.607883,
    "BBD": 2.003905,
    "BDT": 84.18647,
    "BGN": 1.607883,
    "BHD": 0.3741599,
    "BIF": 1960.389,
    "BMD": 1,
    "BND": 1.321357,
    "BOB": 6.84972,
    "BRL": 5.236512,
    "BSD": 0.9934979,
    "BTN": 72.98297,
    "BWP": 10.78774,
    "BYN": 2.514624,
    "BZD": 2.000254,
    "CAD": 1.21333,
    "CDF": 1998,
    "CHF": 0.8865719,
    "CLP": 695.4,
    "CNY": 6.4325,
    "COP": 3775.387,
    "CRC": 611.9086,
    "CUP": 23.81913
  }
}
```

The third screenshot shows the AppBuzz application interface with a 'Best Seller of Week' section. The section displays three books:

- WHILE JUSTICE SLEEPS** by Stacey Abrams. Description: When Justice Wynn slips into a coma, his law clerk, Avery Keene, must unravel the clues of a controversial case. Rank 1 Publisher: Doubleday.
- SOOLEY** by John Grisham. Description: Samuel Sooleyman receives a basketball scholarship to North Carolina Central and determines to bring his family over from a civil war-ravaged South Sudan. Rank 2 Publisher: Doubleday.
- THE LAST THING HE TOLD ME** by John Grisham.

The fourth screenshot shows the browser's developer tools with the Network tab open. The selected request is 'overview.json?api-key=o20RJ...'. The response is a JSON object containing book information:

```
{
  "status": "OK",
  "copyright": "Copyright (c) 2021 The New York Times Company. All Rights Reserved.",
  "copyright": "Copyright (c) 2021 The New York Times Company. All Rights Reserved.",
  "num_results": 99,
  "results": {
    "bestsellers_date": "2021-05-15",
    "published_date": "2021-05-30",
    "list": [
      {
        "list_id": 704,
        "list_name": "Combined Print and E-Book Fiction",
        "list_name_encoded": "combined-print-and-e-book-fiction"
      },
      {
        "list_id": 708,
        "list_name": "Combined Print and E-Book Nonfiction",
        "list_name_encoded": "combined-print-and-e-book-nonfiction"
      },
      {
        "list_id": 1,
        "list_name": "Hardcover Fiction",
        "list_name_encoded": "hardcover-fiction"
      },
      {
        "list_id": 2,
        "list_name": "Hardcover Nonfiction",
        "list_name_encoded": "hardcover-nonfiction"
      },
      {
        "list_id": 17,
        "list_name": "Trade Fiction Paperback",
        "list_name_encoded": "trade-fiction-paperback"
      },
      {
        "list_id": 4,
        "list_name": "Paperback Nonfiction",
        "list_name_encoded": "paperback-nonfiction"
      },
      {
        "list_id": 13,
        "list_name": "Childrens Middle Grade Hardcover",
        "list_name_encoded": "childrens-middle-grade-hardcover"
      },
      {
        "list_id": 7,
        "list_name": "Picture Books",
        "list_name_encoded": "picture-books"
      },
      {
        "list_id": 10,
        "list_name": "Series Books",
        "list_name_encoded": "series-books"
      },
      {
        "list_id": 14,
        "list_name": "Young Adult Hardcover",
        "list_name_encoded": "young-adult-hardcover"
      },
      {
        "list_id": 30,
        "list_name": "Audio Fiction",
        "list_name_encoded": "audio-fiction"
      },
      {
        "list_id": 302,
        "list_name": "Audio Nonfiction",
        "list_name_encoded": "audio-nonfiction"
      },
      {
        "list_id": 532,
        "list_name": "Business Books",
        "list_name_encoded": "business-books"
      },
      {
        "list_id": 719,
        "list_name": "Graphic Books and Manga",
        "list_name_encoded": "graphic-books-and-manga"
      },
      {
        "list_id": 10015,
        "list_name": "Mass Market Monthly",
        "list_name_encoded": "mass-market-monthly"
      },
      {
        "list_id": 10016,
        "list_name": "Middle Grade Paperback Monthly",
        "list_name_encoded": "middle-grade-paperback-monthly"
      },
      {
        "list_id": 10017,
        "list_name": "Young Adult Paperback Monthly",
        "list_name_encoded": "young-adult-paperback-monthly"
      }
    ]
  }
}
```

## Chapter: 6

### 6.0 Product Evaluation

A product evaluation is carried out during any type of project development. It is essential to check for the product requirements that have been assigned during initial planning. The evaluation checks whether the product has been made according to the assigned requirements. Most often, developing a product involves iteration of the same process again and again due to errors during the testing and evaluation phase, During which the developer should return to the previous phase and rectify the errors and begin to develop and again evaluate the product if it does meet the requirements. Product evaluation also shows how well the product can be further improved with change in functionality, design, or by adding other units. The product evaluation process also helps to take a note if any future changes are to be made or not.

In this Advanced Progressive Web Application, the initial proposed project plan was to make an app that contains all the essential features in one application. Also, we wanted to make this application user-friendly by making it easily accessible on mobile devices.

As mentioned in the initial requirement, users are able to access relevant information like news, weather, quotations. For weather, a user is needed to give access to the device location for accurate weather prediction. Users are able to check new books in the list and can buy it from different sites online from this application. Currency rates can be checked and converted as well. For easy and effective communication, users do not require a login process but can directly create or join chat rooms with username and room name.

### 6.1 Introduction

The goal of this chapter was to collect data, the approaches and necessities required for building this progressive web application to promote user's convenience. The exploration of design and development was included throughout the research. For exploration, we did not assign a large set of groups for the survey but we do send this app to technical peers to test the app and send feedback for further improvement. These members from which data was gained to answer the questions, mentioned

this production project implemented the method of investigating the issues, and created a proposal to resolve the issue.

From an application, we modified it to Progressive Web Application, created a separate chat room that doesn't require a login process, and merged all the necessary features like weather, news all in one. Checking out the feedback, we also create user-friendly darkmode as it is the most preferable display mode that users prefer these days. We designed AppBuzz in such a way that it can be accessed offline too but at the same time, it doesn't occupy large storage in the phone. So it is extremely low-end device friendly for users all together.

Hence, these are some of the improvements that we have adhered after proper evaluation of the product through technical peers in our group. Hopefully, this has surely helped AppBuzz to be more effective and user-friendly.

## 6.2 Potential of the project

The Effective use of AppBuzz offers convenience and benefits to users in their daily life as this Web Application is specially designed to provide users features related to their necessity regarding information and communication. Some of the salient features and benefits are highlighted below:

- Up-to-date information

Since AppBuzz mainly focuses on sharing information that needs to be accessed by the users like news, books, weather. Since NYT API is used so, up-to-date and reliable information is sent through the app that keeps users accessible to information easily.

- Available Offline

One of the main features of this Web Application is that it fetches cache on the device so the features of it can be accessed offline too.

- Effective Communication

Appbuzz doesn't require any login processes to communicate with other users. A name and Roomname is enough for this which makes this Web App unique and user friendly.

- Phone friendly

It doesn't occupy a large space in the phone and at the same time it loads very quickly. This Web App is the fusion of Application and Web. AppBuzz easily runs on low-end devices as well.

- All-in-one features

We require different applications to retrieve different features and information. For example: News app for news, Social networking App for communication, e-commerce app for buying, Currency sites, quotation sites, weather prediction app for weather. But Appbuzz has merged all this features in one application so that one application can offer all these features all together

## 6.3 Limitation of the product

Though we have tried our best to improve AppBuzz in the best way possible, still there are some limitations that have arisen. Since, the product is based on APIS so to explore any certain information, users have to visit external links. In the real chat time component, only text and emojis are emphasized. Also, when we install the Application on the phone, it takes a little more time to open. Since it was recently made, it can be accessed through links only. It is not available in play store or can be accessed by searching through search engines.

## Chapter: 7

### 7.0 Project Management and Evaluation

A project evaluation helps to check for whether the project goals and requirements are met. The evaluation discusses the project development and future implementation of different features to make the system better. The future of the project is defined by the status of the current project.

The project began first by creating an initial project plan. The initial plan discussed all the requirements and goals to be achieved from this project. Followed by the initial plan, a product design was created which acted as blueprint for the overall system. The use of Waterfall methodology in this project helped to complete the task stepwise. The development of the product was done adapting the changes made in requirements such as adding most important features and disengaging less required features. The project design phase was followed by the product development phase where a product was created. Apart from product development phase, use of various external tools was used such as use of Office Timeline site and google drive for storing of project files. Officetimeline helps to navigate the progress of the overall project by creating Gantt Chart.. Gantt chart was frequently improvised after the feedback from tutor meetings.

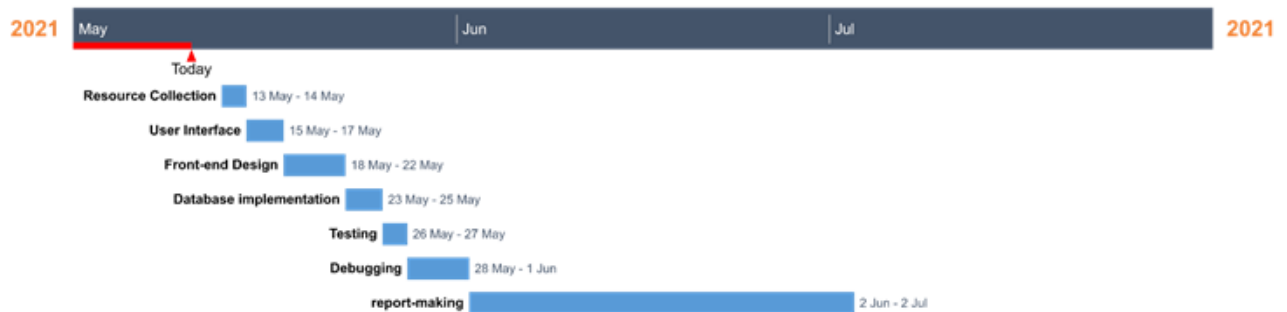


## 7.2 Value of Project Management

Through appropriate project management, we can pledge that the idea and goals of the project are continued and support the spectator's responsibilities and purposes. Moreover, we avoid risks successfully and professionally and use our accessible resources. It similarly supports recognizing the tasks, the deliverables projected, and the timetable needed to complete the project on exact time.

## 7.3 Applied Gantt Chart

Gantt charts are essential project management tools used for planning and scheduling. They collect your tasks in one place on a timeline. From there, you can link dependencies, set milestones, manage resources and more.



Starting date: 13th may,2021, Ending date: 2nd july,2021

## Chapter 7

### 7.0 Summary

Main goal of the project is to provide variety of features needed in day-to-day lives like weather, news, books, currency (currency rates and converter), quotes and chat (a real-time chat) and make features accessible directly from browser by adding them to Home Screen device like a typical native app, skipping app marketplaces and saving valuable storage, especially on low-end devices.

### 7.1 Conclusion

The project was successfully completed analyzing requirements and project plan. The Project was done as the sole Advanced Progressive Web Application for community knowledge sharing. The product development followed Waterfall methodology. Testing on various units of the product showed the product was genuine in its functionality and requirements. Gantt Chart was used to view the progress of the system.

The end product is An Advanced Progressive Web Application - AppBuzz. This Web Application provides variety of features needed in day-to-day lives like weather, news, books, currency (currency rates and converter), quotes and chat (a real-time chat) and make features accessible directly from browser by adding them to Home Screen device like a typical native app, skipping app marketplaces and saving valuable storage, especially on low-end devices.

Overall, the project was successful and every objective and goals were met. The project did teach some valuable information about requirements for any product, Waterfall and other methodologies, use of correct SDLC, importance of various tools and techniques, benefits of storage sites like google, and importance of PWA.

## Appendices

### Appendices I: Initial Proposal

#### BSc (Hons) Computing Course 2020/21 Level 6 Production Project

Name:

Student I.D.:

Course: BSc (Hons) Computing

Supervisor's Name:

#### Final Project Individual Aim & Objectives

**Title of my Project:** An advanced progressive web application - AppBuzz

**Aim of my Project:** The main aim of the project is to build a web application called AppBuzz combining features of both web pages and mobile app.

**Objectives of my Project:** Objectives of my projects are:

1. Provide a variety of features needed in day-to-day lives like weather, news, books, currency (currency rates and converter), quotes and chat (a real-time chat).
2. Make features accessible directly from the browser by adding them to Home Screen devices like a typical native app, skipping app marketplaces and saving valuable storage, especially on low-end devices.

#### Specification of my Product:

1. Product must have used free API.
2. Product should have real-time chat.
3. Product could have a music player.

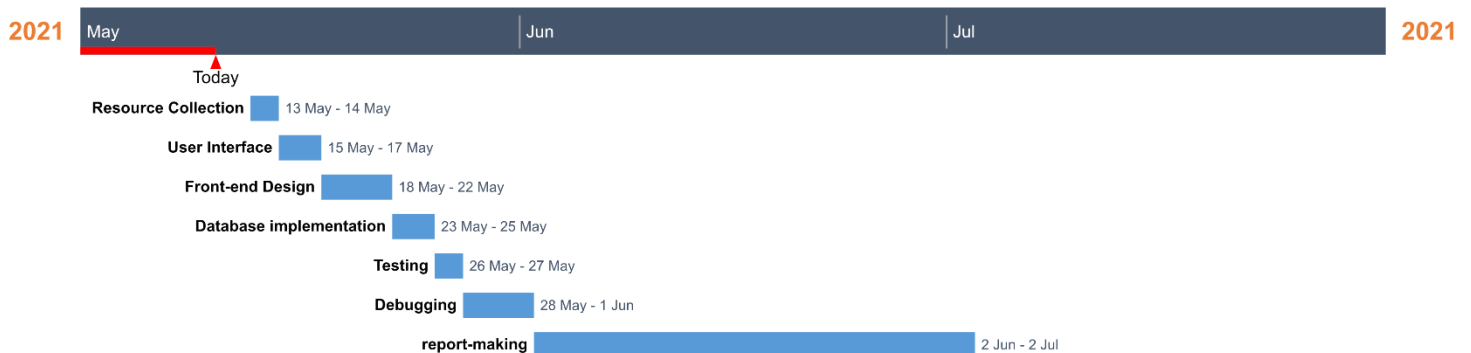
#### 4. Product would have video streaming.

**Research:** For this project, I will do some research and inquiry to access information related to my project. For this, I will visit some relevant websites like W3Schhols, Web.dev and Heroku many more to read blogs, journals. I will study blogs related to progressive web application(PWA), Heroku, node.js, dev.to, journal like " Dawning of progressive web application(PWA)"

**Evaluation:** To plan this project, I have designed a Gantt chart, which will allocate a time limit to each phase of task. I am looking forward to completing this project within the time limit. According to the Gantt chart, I would be designing and developing a live streaming website, which will be a collection of static web pages. If I get sufficient time then I will be hosting my website on a dedicated server along with their suitable domain name.

## Project Planning & Methodology

**Project Planning:** Here, I have created a Gantt Chart for project planning which are highlighted below:



Starting date: 13th may,2021, Ending date: 2nd july,2021

**Methodology:** Since this is my individual project, I will prefer using the waterfall model. The waterfall model is a classical model used in system development life cycle to create a system with a linear and sequential approach. In this model, one phase must have completed to enter another phase. We use an agile model in team work as all the phases start at the same time. Since I am doing this project individually, I use waterfall model.

## Resources

### The hardware and software I require to complete my Project successfully:

To complete this project, I will use various hardware and software tools and I am going to list out all the tools.

**Hardware:** Laptop

**Software:**

1. VScode
2. Xampp Server
3. NPM/node
4. Heroku
5. Browser

## Human Resource

I am working on my Project with the following people


**Name:**

**Role:**

Module Leader

Supervisor

## Initial Bibliography



Maleshkova, Maria. (2010) Investigating web APIs on the World Wide Web:  
University of Bonn.

Jin, Brenda (2018) Designing Web APIs: United States of America :O'Reilly Media Inc.