

ABSTRACT

Description:

M- Commerce is defined as any transaction with monetary value that is conducted via a mobile telecommunication network. M- Commerce like E- Commerce can be B2B (Business to Business), P2P (Person to Person) or B2C (Business to Customer) oriented. Mobile E- Commerce addresses electronic commerce via mobile devices, where the payment procedure is executed via the mobile network. In an E- Commerce site the revenue is the key success metric.

Problem It Solves:

The dashboard is a great way to pull all of this information together, wherein we have the custom report generator along with few basic visualizations i.e: The Mobile Dashboard.

The dashboard will show you the type of customers visiting the site and the sources that bring the quality customers. By the analysis, we can aim for large orders, more profitable or more frequent orders. This dashboard captures all of them so you can analyse and can zero in on which part of the store underperforms. The table with Session Value by channel is to determine where you should be looking for more traffic. The 6 line graphs are to make sure the metrics that determine profit are stable. It compares daily numbers to previous days, previous weeks and previous months.

Questions it should answer:

- Which days of the week have the most purchases?
- Which channels bring visitors that actually buy or do shopping? How profitable are your sales?
- How much do visitors typically order?
- Are the sales improving over the course of time?

Key Metrics:

- Bounce Rate
- Number of visits per page
- Number of sessions
- Sessions by device i.e., web or mobile
- Number of users accessing the website
- Number of visited pages
- Users by time of day
- Sessions by country
- Goal Conversion Rate

Technologies:

Front end: Website (Weebly), Google Demo Account, Java Script (JS)

Back end: GTM (Google Tag Manager), Google Analytics

CONTENTS

DESCRIPTION	PAGE NO
ABSTRACT	i
LIST OF ABBREVIATIONS	ii
LIST OF FIGURES	iii

CHAPTER-1

1. INTRODUCTION	1-8
1.1 Purpose of the Project	2
1.2 Problem with the Existing System	3
1.3 Proposed System	3
1.4 Scope of the Project	4

CHAPTER-2

2. SOFTWARE REQUIREMENTS SPECIFICATIONS	9-12
2.1 Requirements Specification Document	10
2.1.1 Non-Functional Requirements	10
2.1.2 Functional Requirements	11
2.2 Software Requirements	12
2.3 Hardware Requirements	12

CHAPTER-3

3. LITERATURE SURVEY	13-24
-----------------------------	--------------

CHAPTER-4

4. SYSTEM DESIGN	25-40
4.1 Introduction to UML	26
4.2 UML Diagrams	27
4.2.1 Class	diagram
4.2.2	27
4.2.3 Use Case Diagram	29
4.2.4 Activity Diagram	31
Sequence Diagram	34
4.2.5 Component Diagram	37
4.2.6 Deployment Diagram	39

CHAPTER-5

5. IMPLEMENTATION

41-

61

- 5.1 Pseudo code
42
- 5.2 Code Snippets
43

CHAPTER-6

6. TESTING

62-

85

- 6.1 Introduction to Testing
63
- 6.2 Test Cases
85

CHAPTER-7

7. SCREEN SHOTS

86-

92

- 7.1 Home Page
87
- 7.2 Who we are
87
- 7.3 Get in touch.
88
- 7.4 GA Dashboard.
89
- 7.5 Audience Report.
90
- 7.6 Acquisition Report.
91
- 7.7 Behavior Report.
91
- 7.8 GTM Dashboard
92

Conclusion

93

References

94

Bibliography

95

LIST OF ABBRIVATIONS

Acronym	Expansion
GTM	Google Tag Manager
GA	Google Analytics
URL	Universal Resource Locator
JS	Java Script
SRS	Software Requirement Specification
FR	Functional Requirements
NFR	Non- Functional Requirements
CSS	Cascading Style Sheet
HTML	Hyper Text Markup Language
UML	Unified Modelling Language

LIST OF FIGURES

Figure 1.1: Overview of Google Analytics

Figure 1.2: E- Commerce

Figure 1.3: The ABC's

Figure 1.4: Example of E- Commerce Application (Breal)

Figure 3.1: HTML Tags

Figure 3.2: CSS Code

Figure 3.3: Java Script Code

Figure 4.1: Class Diagram for entire application functionality

Figure 4.2: Admin Use case Diagram for entire application functionality

Figure 4.3: User Use case Diagram for entire application functionality

Figure 4.4: Activity Diagram for Google Analytics application

Figure 4.5: Activity Diagram for entire application functionality

Figure 4.6: Sequence Diagram for User

Figure 4.7: Sequence Diagram for Admin

Figure 4.8: Component Diagram for entire application functionality

Figure 4.9: Deployment Diagram for entire application functionality

Figure 6.1: Test Planning

Figure 6.2: Diagram for Automation Testing

Figure 6.3: Bounce Rate per page

Figure 6.4: Number of visits per page

Figure 6.5: Number of sessions

Figure 6.6: Sessions by device

Figure 6.7: Sessions by country

Figure 6.8: Users by time of the day

Figure 6.9: Number of users accessing the website in real time

Figure 6.10: Number of sessions based on date

Figure 7.1: Home Page of the website

Figure 7.2: Journal Page to describe about the website

Figure 7.3: Contact Page

Figure 7.4: Store Page

Figure 7.5: Google Dashboard

Figure 7.6: Audience Report

Figure 7.7: Acquisition Report

Figure 7.8: Behaviour Report

Figure 7.9: Real Time Report

~~Bounce Rate per page~~

CHAPTER– 1

CHAPTER – 1

INTRODUCTION

The online marketing space is in constant shift as new technologies, services, and marketing tactics gain popularity and become the new standard. Online store owners are one of the many different segments affected by these constant evolutions. In order for these business owners to survive and thrive, they need to be able to make better decisions faster. This is where the web analytics i.e., GOOGLE ANALYTICS comes into play.

In a small-to-medium online store, the resources are finite, which means that time and burn rate are critical factors to success. Without knowing which marketing activities are working, we will be wasting both time and money. Another symptom of this lack of information is that we will be missing out on profit from the channels that are working, because instead of doubling down on these channels, your budget will be spread across on both profitable and unprofitable channels.

To explain this let's create a sample website HTML web application using Weebly. Having relevant statistical information on fingertips is the first step in building a foundation for continuous experimentation on this website and other areas of online presence. Here we will be able to test certain copy and the overall layout of the e-commerce site which is the next logical step for a profitable business that wants to raise its profits.

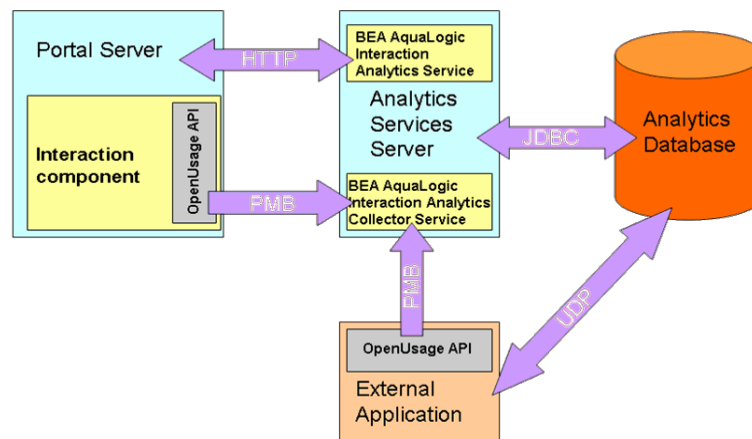


Figure 1.1: Overview of Google Analytics

1.1 Purpose of the Project:

Having access to statistical information from all areas of the online marketing and sales activities gives us an advantage over competitors that do not have this information. Understanding trends and which marketing channels are no longer profitable allows us to manoeuvre as a business before damage is done to your bottom line. Understanding shifts in consumer behaviour gives us insights into the demands of the market. Knowing these things enables us to drop certain products or make strategic changes in the pricing that will result in big gains or, at the very least, limit damage to your profits.

The highlighted aspects that serve the purpose of the project are:

- To analyse and improve the sales of any business
- To improve the performance of the website
- To increase the number of visitors of the website
- To impact the workflow and decision making processes of any business
- To achieve maximum returns of an investment



Figure 1.2: E- Commerce

1.2 Problems with the Existing System:

GA provides us with the actual reports and analytics on how the website visitors behave once on the website, who they are by their age, gender, location, etc., how they landed on your site (traffic source), the most popular content on your site, your total conversions and so on. With this information in hand, we can plan fully informed business strategies and grow the business faster. Imagining things to be done without these provisions is simply impossible for better future prospects.

In the existing system let's look at the aspects where GA plays a crucial role and without which the system has its debts. They are as follows:

- By using traditional methods of taking feedbacks from the customers and surveying isn't helping the cause for finding the products that are high in sale
- Website owners aren't able to work accordingly to user's choices
- Majority of products in a website aren't sold to the potential which affects in revenue of the firm
- The companies cannot know the loop holes and the changes required to improve the website or the product's sales and hence cannot improve one's decision making skills
- The user cannot have the count of number of visitors to his site
- One cannot track the audience information
- Any error in analysing the sales can affect the decisions, which in turn bring down the ranking of the exam
- The system cannot handle the vast amount of data (Big Data) which is to be analysed
- The speed of data creation coupled with advantage gained from analysing the data in real time is not possible

1.3 Proposed System:

In order to evaluate web marketing success it is needed to set measurable and attainable goals. Analyzing trends and results of the online marketing strategy means identifying key performance indicators (KPI) that measure what is happening on the website and the other company's social channels. Web analytics tools like GA allow to collect this data and rely on useful metrics using which we can make analysis of data. GA has become a standard tool when it comes to web analytics because of its ease of use, informative reports, and the fact that it's free. GA is a very powerful tool for E- Commerce sites because Google allows you to send all your sales data to your GA account. Once this integration is set up, all your sales will be tied to actual sessions, allowing you to connect sales to specific marketing channels.

Setting up E- Commerce tracking in GA is a multi-step process which requires to first enable E- Commerce tracking in your GA admin and then make necessary changes to the code (website). Here comes the role of the GTM (Google Tag Manager).

Using GA any user can:

- Create custom reports
- Integrate with other tools
- Show real-time traffic data
- Understand where your traffic came from
- Rank your pages by popularity
- Track campaigns
- Export to Excel

Basically knowing how to navigate the GA interface, understanding what sort of data is natively collected, and having some tactical skills in place for how to dig into it is important. This lesson is going to focus on some of the basics of GA with the goal of giving you some skills that allow you better understand your web traffic. Google has built into their platform a framework to help users understand analytics better; The “ABCs” of Acquisition, Behaviour & Conversion. The default view of many reports in GA breaks up metrics into these three categories.

- Acquisition= How do you acquire users? How many? From where?
- Behaviour= How do users behave on your site?
- Conversion= Do users take a desired action on the site?

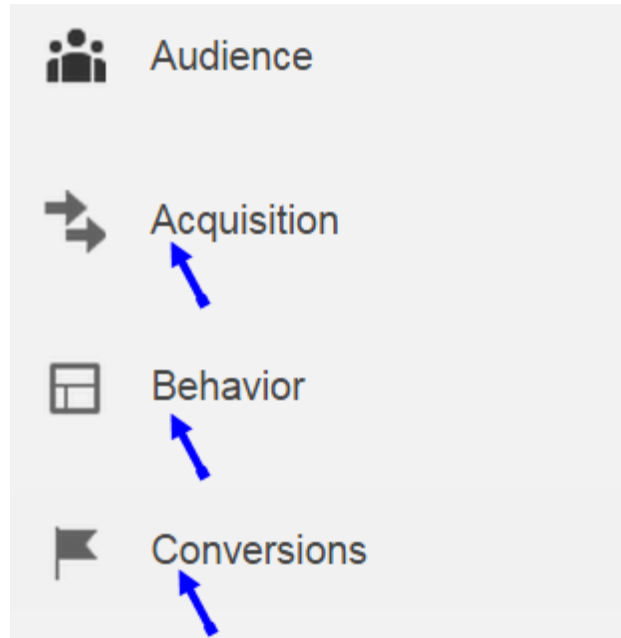


Figure 1.3: The ABC's

1.4 Scope of the System:

The Mobile E- Commerce using Google Analytics can be clearly analysed through Google Analytics. This project is used to nurture the needs of the user in the E- Commerce sector by embedding all sales taking place in an E- Commerce site. It will also help optimize the content of the business's website based on the user's interest, which can be known by measuring the results of the website in real time, compare the data to previous periods, and other techniques can also be used. It will provide both speed and accuracy to business decisions. It provides answers to present business problems as well as give a view of future trends.

Google Analytics: Google Analytics is a premium web analytics service offered by Google that tracks and reports website traffic, currently as a platform inside the Google Marketing Platform brand. Google launched the service in November 2005 after acquiring Urchin. GA is now the most widely used web analytics service on the web. Google Analytics also provides an SDK that allows gathering usage data from iOS and Android Apps, known as Google Analytics for Mobile Apps.

Features: Integrated with Google Ad Words, users can now review online campaigns by tracking landing page quality and conversions (goals). Goals might include sales, lead generation, viewing a specific page, or downloading a particular file. Google Analytics

Mobile E- Commerce using Google Analytics

approach is to show high-level, dashboard-type data for the casual user, and more in-depth data further into the report set. Google Analytics analysis can identify poorly performing pages with techniques such as funnel visualization, where visitors came from (referrers), how long they stayed on the website and their geographical position. It also provides more advanced features, including custom visitor segmentation.

Google Analytics e-commerce reporting can track sales activity and performance. The e-commerce reports show a site's transactions, revenue, and many other commerce-related metrics.

On September 29, 2011, Google Analytics launched Real Time analytics, enabling a user to have insight about visitors currently on the site. A user can have 100 site profiles. Each profile generally corresponds to one website. It is limited to sites which have traffic of fewer than 5 million page views per month (roughly 2 page views per second) unless the site is linked to an Ad Words campaign. Google Analytics includes Google Website Optimizer, rebranded as *Google Analytics Content Experiments*.

Google Analytics' Cohort analysis feature helps understand the behaviour of component groups of users apart from your user population. It is beneficial to marketers and analysts for successful implementation of a marketing strategy.

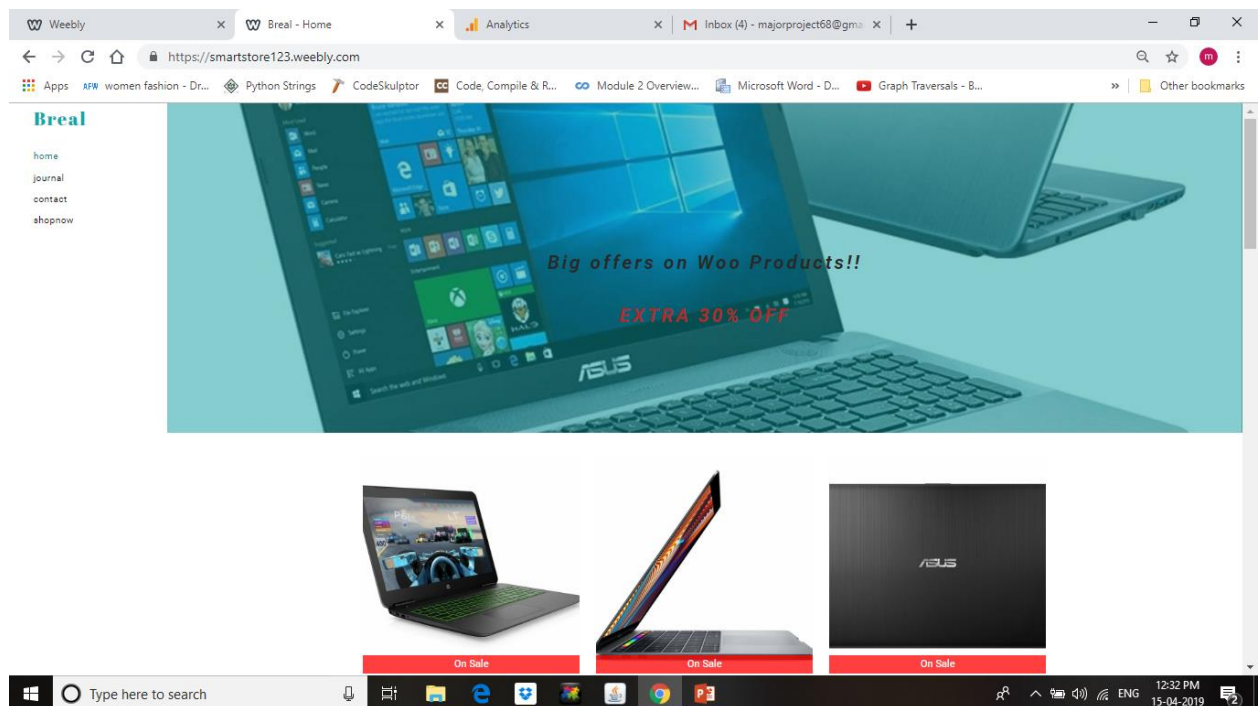


Figure 1.4: Example of E- Commerce Application (Breal)

CHAPTER- 2

CHAPTER- 2

SOFTWARE REQUIREMENTS SPECIFICATION

2.1 Requirements Specification Document:

The purpose of SRS (Software Requirement Specification) document is to describe the external behaviour of the application developed or software. It defines the operations, performance and interfaces and quality assurance requirement of the application or software. The complete software requirements for the system are captured by the SRS. This section introduces the requirement specification document for Mobile E- Commerce which enlists functional as well as non-functional requirements.

2.1.1 Functional Requirements:

For documenting the functional requirements, the set of functionalities supported by the system are to be specified. A function can be specified by identifying the state at which data is to be input to the system, its input data domain, the output domain, and the type of processing to be carried on the input data to obtain the output data.

Functional requirements define specific behaviour or function of the application. Following are the functional requirements:

- 2. 1. 1. 1. Bounce Rate
- 2. 1. 1. 2. Number of visits per page
- 2. 1. 1. 3. Number of sessions
- 2. 1. 1. 4. Sessions by device i.e., web or mobile
- 2. 1. 1. 5. Number of users accessing the website
- 2. 1. 1. 6. Number of visited pages
- 2. 1. 1. 7. Users by time of day
- 2. 1. 1. 8. Sessions by country
- 2. 1. 1. 9. Goal Conversion Rate

2.1.2 Non-Functional Requirements:

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. Especially these are the constraints the system must work within. Following are the non-functional requirements:

- 2. 1. 1. 1. The website needs to be fast and responsive
- 2. 1. 1. 2. The website needs to have a smooth checkout flow and it must be personalized
- 2. 1. 1. 3. Searching for product using name and category
- 2. 1. 1. 4. Adding products to the cart
- 2. 1. 1. 5. Being able to make payment successfully
- 2. 1. 1. 6. Getting the acknowledgement page with positive response

2.3 Software Requirements:

- 2. 3. 1 Operating System : Windows 7/8/10, Linux and iOS
- 2. 3. 2 Platform (Analytics Tool) : Google Analytics
- 2. 3. 3 Programming Language : Website, Google Demo Account, JavaScript
- 2. 3. 4 Tag Manager : GTM
- 2. 3. 5 Browser : Google Chrome, Mozilla Firefox, Safari and Internet Explorer
- 2. 3. 6 Internet

2.4 Hardware Requirements:

2. 4. 1 Processor	:	Intel Pentium or Intel Celeron
2. 4. 2 Hard Disk	:	40 GB or more
2. 4. 3 RAM	:	512 MB or more
2. 4. 4 Cache Memory	:	100MB

CHAPTER- 3

CHAPTER- 3

LITREATURE SURVEY

Software Development Life Cycle (SDLC):

The software development life cycle (SDLC) is a framework defining tasks performed at each step in the software development process. SDLC is a structure followed by a development team within the software organization. It consists of a detailed plan describing how to develop, maintain and replace specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

The software development life cycle is also known as the software development process.

SDLC consists of following activities:

1. **Planning:** The most important parts of software development, requirement gathering or requirement analysis are usually done by the most skilled and experienced software engineers in the organization. After the requirements are gathered from the client, a scope document is created in which the scope of the project is determined and documented.
2. **Implementation:** The software engineers start writing the code according to the client's requirements.
3. **Testing:** This is the process of finding defects or bugs in the created software.
4. **Documentation:** Every step in the project is documented for future reference and for the improvement of the software in the development process. The design documentation may include writing the application programming interface (API).
5. **Deployment and maintenance:** The software is deployed after it has been approved for release.
6. **Maintaining:** Software maintenance is done for future reference. Software improvement and new requirements (change requests) can take longer than the time needed to create the initial development of the software.

Technologies Used:

1. Hyper Text Markup Language (HTML):

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of Cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the Appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the Rendered page. HTML provides a means to create structured documents by denoting structural Semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML Elements are delineated by tags, written using angle brackets. Tags such as `` and `<input/>` Directly introduce content into the page.

Browsers do not display the HTML tags, but use them to interpret the content of the page. HTML can embed the programs written in a scripting language such as JavaScript, which affects the behaviour and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

Sample Code:

```
<!DOCTYPE html><html>
```

```
<head>
```

```
<title>Page Title</title>
```

```
</head>
```

Mobile E- Commerce using Google Analytics

<body>

<h1>This is a Heading</h1>

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

</body>

</html>

Tag	Usage
 	Code for a nonbreaking space
<!-- -->	Comments
	Creating links
	Bold Text
<basefont size= 1> </basefont>	Set base font size from 1 to 7
<blockquote></blockquote>	Separates text
<body></body>	Body of HTML
 </br>	Line break
<button ...></button>	Adds a button
<caption></caption>	Adds table caption
<center></center>	Centers text
<div></div>	Division of code
	Adds emphasis or bolding
	Selects font
<frameset></frameset>	Start of frames
<h1></h1>	Heading 1, which is the largest heading size normally used for titles
<h2></h2>	Heading 2
<h3></h3>	Heading 3
<h4></h4>	Heading 4
<h5></h5>	Heading 5
<h6></h6>	Heading 6
<head></head>	Head of HTML document
<hr></hr>	Horizontal line
<html></html>	Start of any HTML document
<i></i>	Italics
<iframe></iframe>	An inline frame

Tag	Usage
	Image source
<input name></input>	Input form
	List item
<meta name= ></meta>	Meta information
	Numbered list
<option value="list"></option>	Option form
<p></p>	New paragraph
<param></param>	Sets a parameter on an element
<q></q>	Quotation
<select name></select>	Selection Form
<strike></strike>	Strikethrough text
<style></style>	Style sheet definition
	Subscript text
	Superscript text
<table></table>	Table definition
<td></td>	Data cell definition
<th></th>	Header cell for a table
<title></title>	Document title
<tr></tr>	Table row
	Bulleted list

Figure 3.1: HTML Tags

2. Cascading Style Sheets:

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World

Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name *cascading* comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable. The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

In addition to HTML, other markup languages support the use of CSS, including XHTML, plain XML, SVG, and XUL.

Sample Code:

```
h1 { color: white;
background: orange;
border: 1px solid black
padding: 0 0 0 0;
font-weight: bold;
}
/* begin: seaside-theme */

body {
background-color:white;
color:black;
font-family:Arial,sans-serif;
margin: 0 4px 0 0;
border: 12px solid;
}
```

Figure 3.2: CSS code

3. Java Script:

JavaScript often abbreviated as **JS**, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm. Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has an API for working with text, arrays, dates, regular expressions, and basic manipulation of the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments

that makes JavaScript available for writing mobile and desktop applications, including desktop widgets. Although there are strong outward similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design; JavaScript was influenced by programming languages such as Self and Scheme.

Sample Code:

```
JavaScript

function retrieveWebSite(siteUrl) {
    var clientContext = new SP.ClientContext(siteUrl);
    this.oWebsite = clientContext.get_web();

    clientContext.load(this.oWebsite);

    clientContext.executeQueryAsync(
        Function.createDelegate(this, this.onQuerySucceeded),
        Function.createDelegate(this, this.onQueryFailed)
    );
}

function onQuerySucceeded(sender, args) {
    alert('Title: ' + this.oWebsite.get_title() +
        ' Description: ' + this.oWebsite.get_description());
}

function onQueryFailed(sender, args) {
    alert('Request failed. ' + args.get_message() +
        '\n' + args.get_stackTrace());
}
```

Figure 3.3: JavaScript code

4. Google Analytics:

Google Analytics is a premium web analytics service offered by Google that tracks and reports website traffic, currently as a platform inside the Google Marketing Platform brand. Google launched the service in November 2005 after acquiring Urchin. Google Analytics is now the most widely used web analytics service on the web. Google Analytics also provides an SDK that allows gathering usage data from iOS and Android Apps, known as *Google Analytics for Mobile Apps*.

Features:

Integrated with Google Ad Words, users can now review online campaigns by tracking landing page quality and conversions (goals). Goals might include sales, lead generation, viewing a specific page, or downloading a particular file. Google Analytics' approach is to show high-level, dashboard-type data for the casual user, and more in-depth data further into the report set. Google Analytics analysis can identify poorly performing pages with techniques such as funnel visualization, where visitors came from (referrers), how long they stayed on the website and their geographical position. It also provides more advanced features, including custom visitor segmentation. Google Analytics e-commerce reporting can track sales activity and performance.

The e-commerce reports show a site's transactions, revenue, and many other commerce-related metrics.

On September 29, 2011, Google Analytics launched Real Time analytics, enabling a user to have insight about visitors currently on the site. A user can have 100 site profiles. Each profile generally corresponds to one website. It is limited to sites which have traffic of fewer than 5 million page views per month (roughly 2 page views per second) unless the site is linked to an Ad Words campaign. Google Analytics includes Google Website Optimizer, rebranded as *Google Analytics Content Experiments*. Google Analytics' Cohort analysis feature helps understand the behaviour of component groups of users apart from your user population. It is beneficial to marketers and analysts for successful implementation of a marketing strategy.

Technology:

Google Analytics is implemented with "page tags", in this case, called the Google Analytics Tracking Code, which is a snippet of JavaScript code that the website owner adds to every page of the website. The tracking code runs in the client browser when the client browses the page (if JavaScript is enabled in the browser) and collects visitor data and sends it to a

Google data collection server as part of a request for a web beacon.

The tracking code loads a larger JavaScript file from the Google web server and then sets variables with the user's account number. The larger file (currently known as ga.js) was typically 18 KB, but the date of the file when it was that size is unknown. However, the more recent size is over 40KBytes as of May 2018. A "c: dir. ga.js /s" command from a PC shows the following file sizes between 2009 ~ 2018 (2009/10/07 23,536 ga.js, 2014/05/27 40,155 ga.js, 2014/10/10 40,924 ga.js, 2015/09/13 41,100 ga.js, 2017/08/06 43,082 ga.js, 2018/04/08 46,275 ga.js).

The file does not usually have to be loaded, however, due to browser caching. Assuming caching is enabled in the browser, it downloads ga.js only once at the start of the visit. Furthermore, as all websites that implement Google Analytics with the ga.js code use the same master file from Google, a browser that has previously visited any other website running Google Analytics will already have the file cached on their machine.

In addition to transmitting information to a Google server, the tracking code sets a first party cookie (If cookies are enabled in the browser) on each visitor's computer. This cookie stores anonymous information called the Client Id. Before the launch of Universal Analytics, there were several cookies storing information such as whether the visitor had been to the site before.

Google Analytics Tracking Code:

```
<!-- Global site tag (gtag.js) - Google Analytics -->
```

```
<script async src="https://www.googletagmanager.com/gtag/js?id=UA-134249147-1"></script>
```

```
<script>
```

```
  window.dataLayer = window.dataLayer || [];
```

```
  function gtag(){dataLayer.push(arguments);}
```

```
  gtag('js', new Date());
```

```
gtag('config', 'UA-134249147-1');  
  
</script>
```

Google Tag Manager:

Google Tag Manager is a free tool that allows you manage and deploy marketing tags (snippets of code or tracking pixels) on your website (or mobile app) without having to modify the code. Information from one data source (your website) is shared with another data source (Analytics) through Google Tag Manager. GTM becomes very handy when you have lots of tags to manage because all of the code is stored in one place. A huge benefit of Tag Manager is that you, the marketer, can manage the code on your own. There are three main parts to Google Tag Manager:

Tags: Snippets of Javascript or tracking pixels

- **Triggers:** This tells GTM when or how to fire a tag
- **Variables:** Additional information GTM may need for the tag and trigger to work

What are tags?

Tags are snippets of code or tracking pixels from third-party tools. These tags tell Google Tag Manager *what* to do.

Examples of common tags within Google Tag Manager are:

- Google Analytics Universal tracking code
- Adwords Remarketing code
- Adwords Conversion Tracking code
- Heatmap tracking code (Hotjar, CrazyEgg, etc...)
- Facebook pixels

What are triggers?

Triggers are a way to fire the tag that you set up. They tell Tag Manager *when* to do what you want it to do.

What are variables?

Variables are additional information that GTM *may* need for your tag and trigger to work. The most basic type of constant variable that you can create in GTM is the Google Analytics UA number (the tracking ID number).Google Tag Manager is a completely different tool used only for storing and managing third-party code. There are no reports or any way to do analysis in GTM.

Google Analytics is used for actual reporting and analysis. All conversion tracking goals or filters are managed through Analytics.All reporting (conversion reports, custom segments, ecommerce sales, time on page, bounce rate, engagement reports, etc...) are done in Google Analytics.

Benefits of Google Tag Manager:

Once you get over the learning curve, what you can do in Google Tag Manager is pretty amazing. You can customize the data that is sent to Analytics.

You can setup and track basic events like PDF downloads, outbound links or button clicks. Or, complex enhanced ecommerce product and promotion tracking.

Let's say we want to track all outbound links on the website. In GTM, choose the category name, action and label. We chose offsite link, click and click URL.

In Google Analytics go to Behavior > Events > Top Events > Offsite link.

Now select either event action or label to get the full reports. The data that we setup in Google Tag Manager is now appearing in the Analytics reports.

CHAPTER- 4

CHAPTER- 4

SYSTEM DESIGN

4.1 Introduction to UML:

The Unified Modelling Language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic, semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows:

1. User Model View

- i. This view represents the system from the users' perspective.
- ii. The analysis representation describes a usage scenario from the end-users' perspective.

2. Structural Model View

- i. In this model, the data and functionality are arrived from inside the system.
- ii. This model view models the static structures.

3. Behavioural Model View

It represents the dynamic of behavioural as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

4. Implementation Model View

In this view, the structural and behavioural as parts of the system are represented as they are to be built.

5. Environmental Model View

In this view, the structural and behavioural aspects of the environment in which the system is to be implemented are represented.

4.2 UML Diagrams:

4. 2. 1 Class Diagram:

In software engineering, a **class diagram** in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

The class diagram is the main building block of object-oriented modelling. It is used for general conceptual modelling of the structure of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modelling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

In the diagram, classes are represented with boxes that contain three compartments:

- The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.
- The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.
- The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

A Class with three compartments:

In the design of a system, a number of classes are identified and grouped together in a class diagram that helps to determine the static relations between them. With detailed modelling, the classes of the conceptual design are often split into a number of subclasses.

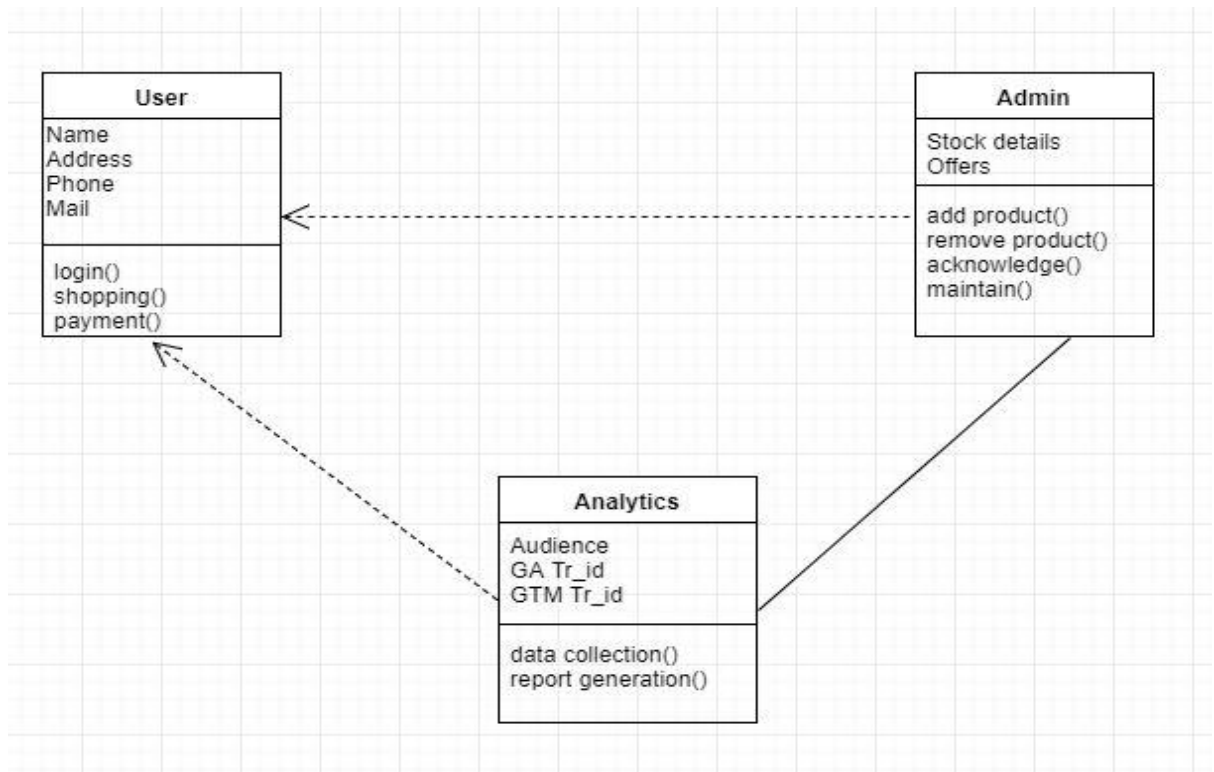


Figure 4.1: Class Diagram for entire application functionality

4.2.2 Use Case Diagram:

To model a system, the most important aspect is to capture the dynamic behaviour. To clarify a bit in details, dynamic behaviour means the behaviour of the system when it is running /operating.

So only static behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. So use case diagrams are consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system. So to model the entire system numbers of use case diagrams are used.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analysed to gather its functionalities use cases are prepared and actors are identified.

In brief, the purposes of use case diagrams can be as follows:

- a. Used to gather requirements of a system.
- b. Used to get an outside view of a system.
- c. Identify external and internal factors influencing the system.
- d. Show the interacting among the requirements are actors.

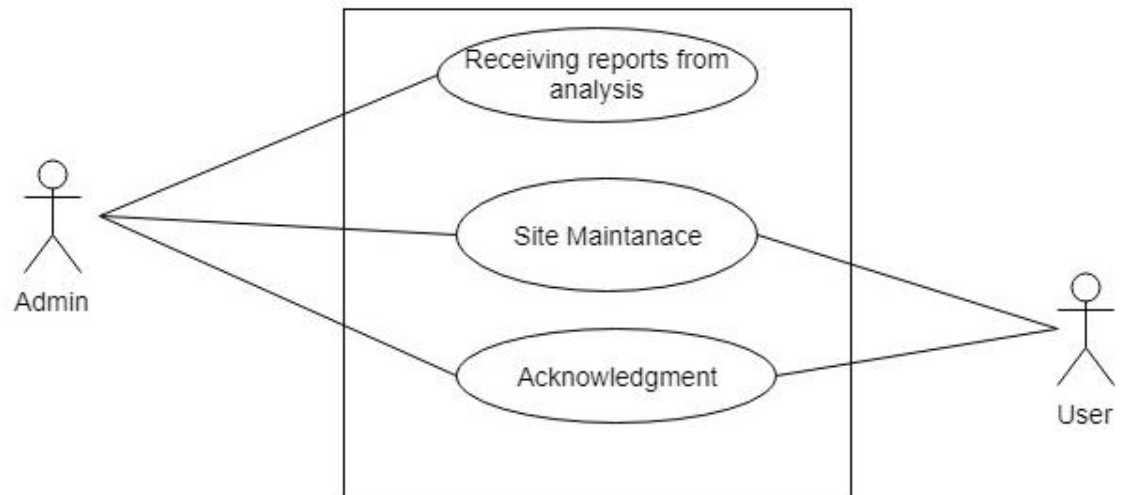


Figure 4.2: Admin Use case Diagram for entire application functionality

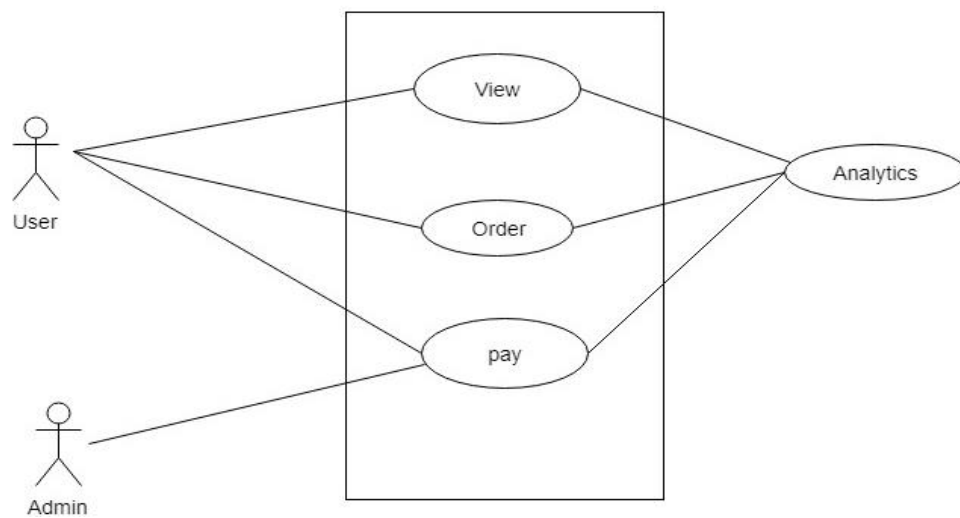


Figure 4.3: User Use case Diagram for entire application functionality

4.2.3 Activity Diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities.

Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores.

Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most important shape types:

- ellipses represent actions
- diamonds represent decisions
- bars represent the start (split) or end (join) of concurrent activities
- a black circle represents the start (initial node) of the workflow
- an encircled black circle represents the end (final node)

Arrows run from the start towards the end and represent the order in which activities happen.

Activity diagrams can be regarded as a form of a structured flowchart combined with a traditional data flow diagram.

Typical flowchart techniques lack constructs for expressing concurrency.

However, the join and split symbols in activity diagrams only resolve this for simple cases; the meaning of the model is not clear when they are arbitrarily combined with decisions or loops.

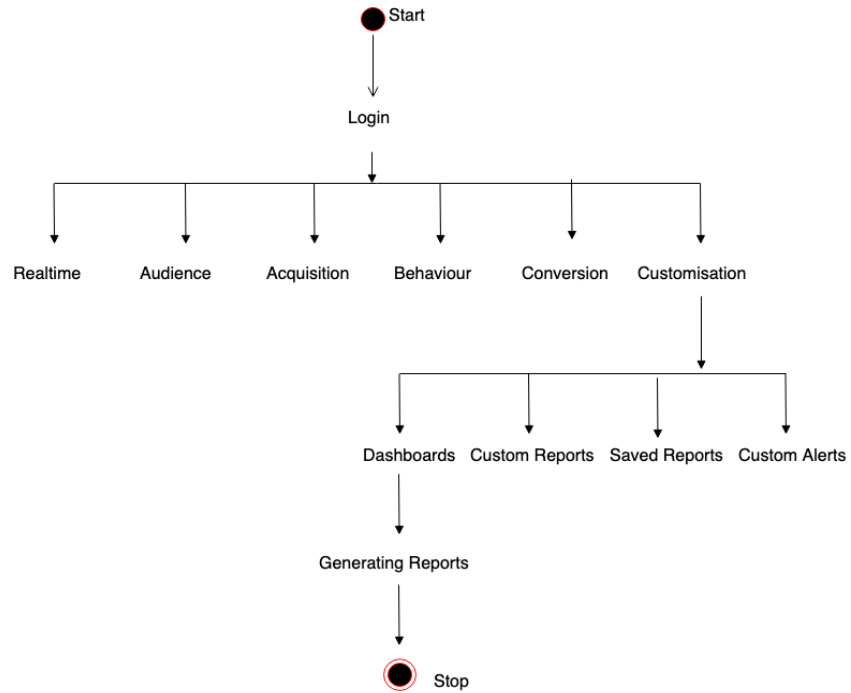


Figure 4.4: Activity Diagram for Google Analytics application functionality

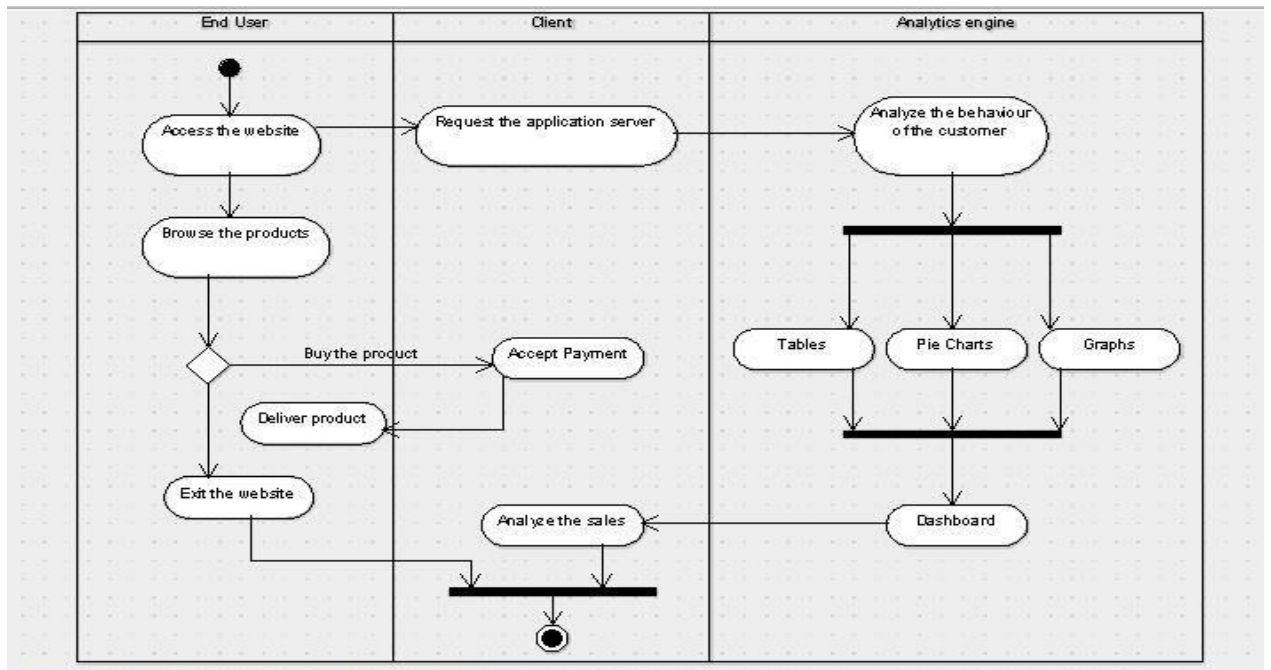


Figure 4.5: Activity Diagram for entire application functionality

4.2.4 Sequence Diagram:

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modelling a new system.

The aim of a sequence diagram is to define event sequences, which would have a desired outcome. The focus is more on the order in which messages occur than on the message per se. However, the majority of sequence diagrams will communicate what messages are sent and the order in which they tend to occur.

Basic Sequence Diagram Notations are:

- **Class Roles or Participants**

Class roles describe the way an object will behave in context. Use the UML object symbol to illustrate class roles, but don't list object attributes.

- **Activation or Execution Occurrence**

Activation boxes represent the time an object needs to complete a task. When an object is busy executing a process or waiting for a reply message, use a thin grey rectangle placed vertically on its lifeline.

- **Messages**

Messages are arrows that represent communication between objects. Use half-arrowed lines to represent asynchronous messages.

Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks.

- **Lifelines**

Lifelines are vertical dashed lines that indicate the object's presence over time.

- **Destroying Objects**

Objects can be terminated early using an arrow labelled "<< destroy >>" that points to an X. This object is removed from memory. When that object's lifeline ends, you can place an X at the end of its lifeline to denote a destruction occurrence.

- **Loops**

A repetition or loop within a sequence diagram is depicted as a rectangle. Place the condition for exiting the loop at the bottom left corner in square brackets [].

- **Guards**

When modelling object interactions, there will be times when a condition must be met for a message to be sent to an object. Guards are conditions that need to be used throughout UML diagrams to control flow.

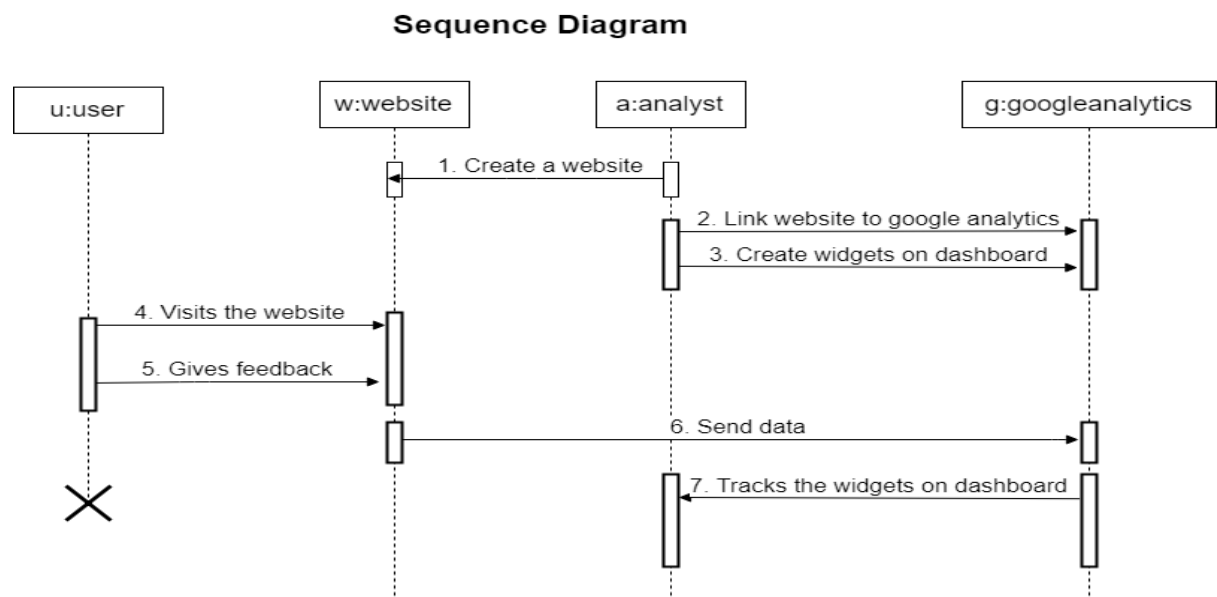


Figure 4.6: Sequence Diagram for User

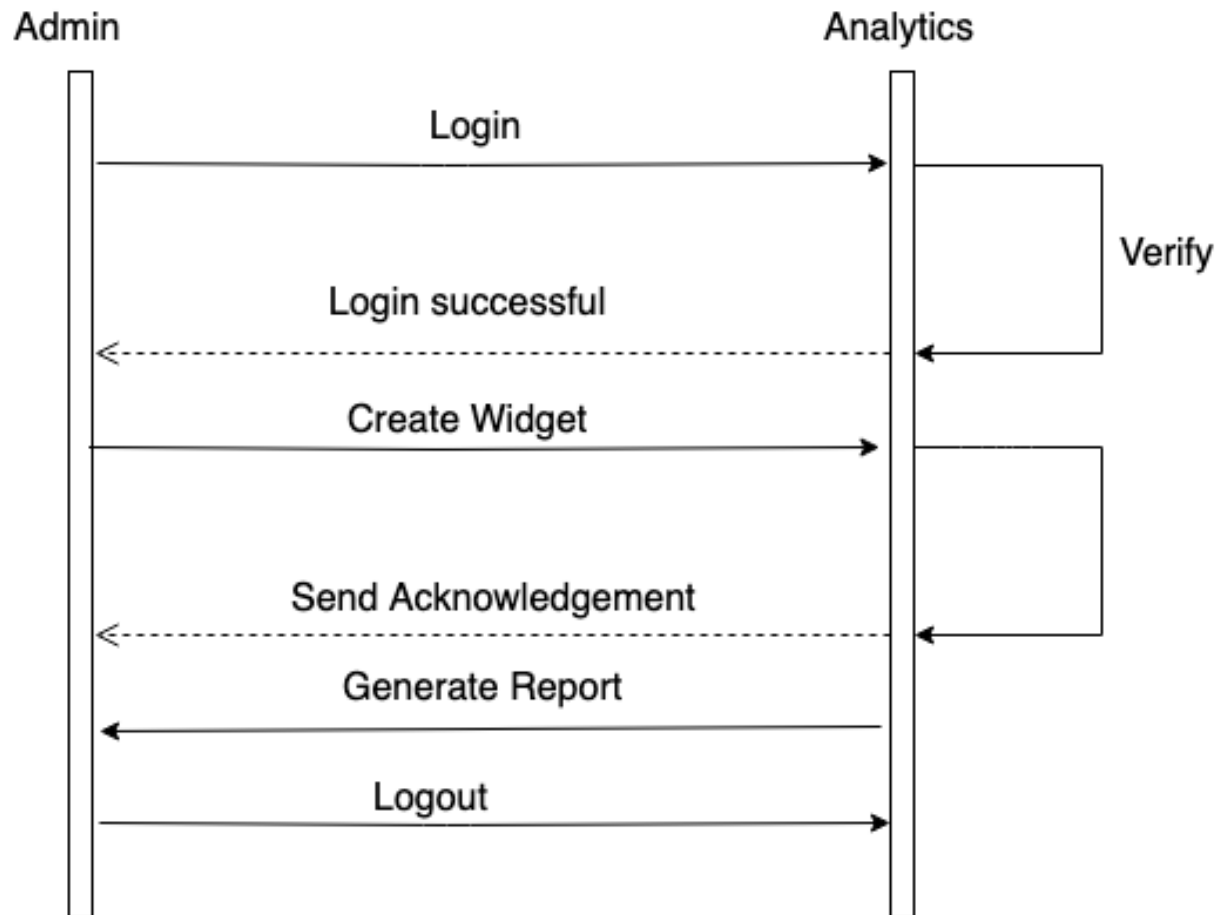


Figure 4.6: Sequence Diagram for Admin

4.2.5 Component Diagram:

A component is something required to execute a stereotype function. Examples of stereotypes in components include executable's, documents, database tables, files, and library files. Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.

An assembly connector is a "connector between two components that defines that one component provides the services that another component requires. An assembly connector is a connector that is defined from a required interface or port to a provided interface or port."

When using a component diagram to show the internal structure of a component, the provided and required interfaces of the encompassing component can delegate to the corresponding interfaces of the contained components. A delegation connector is a "connector that links the external contract of a component (as specified by its ports) to the internal realization of that behaviour by the component's parts."

The example above illustrates what a typical insurance policy administration system might look like. Each of the components depicted in the above diagram may have other component diagrams illustrating its internal structure.

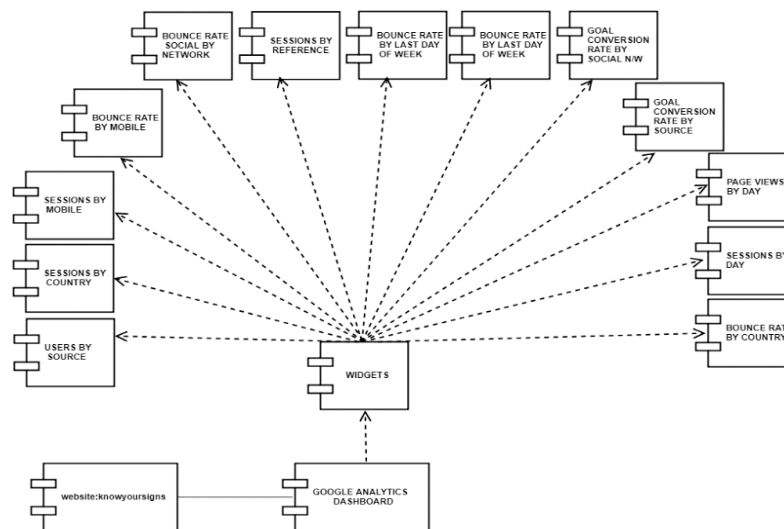


Figure 4.8: Component Diagram for entire application functionality

4.2.6 Deployment Diagram:

A **deployment diagram** in the Unified Modelling Language models the physical deployment of artefacts on nodes. To describe a website, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artefacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

The nodes appear as boxes, and the artefacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

There are two types of nodes. They are:

1. Device Node
2. Execution Environment Node

Device nodes are physical computing resources with processing memory and services to execute software, such as typical computers or mobile phones. An execution environment node (EEN) is a software computing resource that runs within an outer node and which itself provides a service to host and execute other executable software elements.

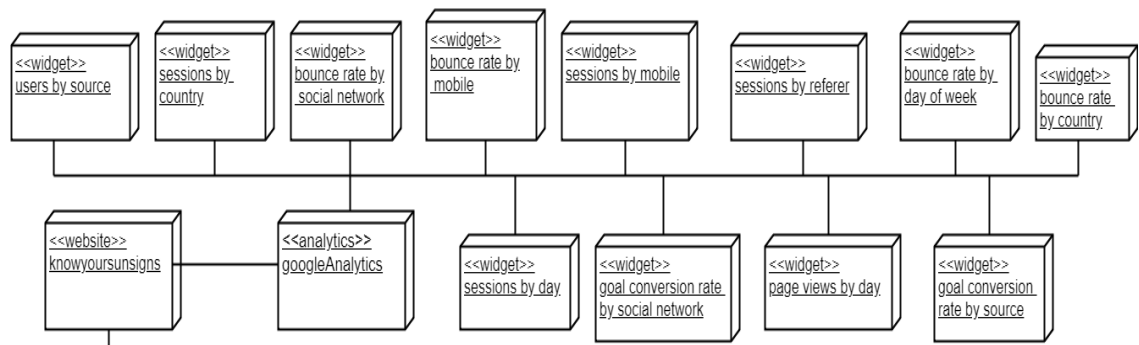


Figure 4.9: Deployment Diagram for entire application functionality

CHAPTER- 5

CHAPTER- 5

IMPLEMENTATION

5.1 Pseudo code:

Step 1:

Open the browser and type www.weebly.com

Step 2:

If you already have account, then Log in or else Sign Up if you want to create a new account.

Step 3:

You can create a new account by filling up the given form with you details such as mail id, full name and password or you can login through either Facebook or Gmail.

Step 4:

After creating the Weebly account you can choose a theme and domain name related to your project and start building it with the various tools available in the Weebly.

Ex: Images, buttons, products, product categories etc.

Step 5:

After building the website completely click on publish button and your website link will be displayed.

Step 6:

Now for retrieving the data from the website we should link Google analytics to Weebly. So create a new Google analytics account and link it to your website by pasting the header code of the Google analytics in the SEO setting of the Weebly.

Step 7:

To track the events and create triggers and tags create a Google tag manager account and paste the header code of it right above the header code of Google analytics in the Weebly SEO settings and footer code in the embedded code on the desired page.

Step 8:

Now both tag manager and Google analytics are connected to Weebly. Now you can track the events and transactions.

5.2 Code Snippets:

Home Page:

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
    <head><script  
src="/gdpr/gdprscript.js?buildTime=1555696067&hasRemindMe=true&stealth=false"></script>
```

```
    <title>Breal - Home</title><meta property="og:site_name" content="Breal" />
```

```
<meta property="og:title" content="Breal" />
```

```
<meta property="og:description" content="Breal" />

<meta property="og:image"
content="https://smartstore123.weebly.com/uploads/1/2/4/8/124851663/edited/61uojnxvbpl-
ac-ul320-sr170-320.jpg?1555249166" />

<meta property="og:url" content="https://smartstore123.weebly.com/" />

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<link id="wsite-base-style" rel="stylesheet" type="text/css"
href="//cdn2.editmysite.com/css/sites.css?buildTime=1555696067" />

<link rel="stylesheet" type="text/css"
href="//cdn2.editmysite.com/css/old/fancybox.css?1555696067" />

<link rel="stylesheet" type="text/css" href="//cdn2.editmysite.com/css/social-
icons.css?buildtime=1555696067" media="screen,projection" />

<link rel="stylesheet" type="text/css" href="/files/main_style.css?1555928422" title="wsite-
theme-css" />

<link
href="//fonts.googleapis.com/css?family=Roboto:400,300,300italic,700,400italic,700italic&s
ubset=latin,latin-ext" rel='stylesheet' type='text/css' />

<link href="//fonts.googleapis.com/css?family=Droid+Sans:400,700&subset=latin,latin-ext"
rel='stylesheet' type='text/css' />

<link href="//fonts.googleapis.com/css?family=Abril+Fatface&subset=latin,latin-ext"
rel='stylesheet' type='text/css' />

<style type='text/css'>

<script>
```



```
var STATIC_BASE = '//cdn1.editmysite.com/';

var ASSETS_BASE = '//cdn2.editmysite.com/';

var STYLE_PREFIX = 'wsite';

</script>

<script src='https://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js'></script>

<script type="text/javascript"
src="//cdn2.editmysite.com/js/lang/en/stl.js?buildTime=1555696067"></script>

<script
src="//cdn2.editmysite.com/js/site/main.js?buildTime=1555696067"></script><script
type="text/javascript">

</script>

<!-- Global site tag (gtag.js) - Google Analytics -->

<script async src="https://www.googletagmanager.com/gtag/js?id=UA-138251933-1"></script>

<script>

    window.dataLayer = window.dataLayer || [];

    function gtag(){dataLayer.push(arguments);}

    gtag('js', new Date());

    gtag('config', 'UA-138251933-1');

</script>
```

```
</head>
```

```
<body class="header-page wsite-page-index nav-position-sidebar sticky-nav-on full-width-  
nav-on full-width-body-off header-overlay-on wsite-theme-light"><div class="header">
```

```
<div class="nav-wrap">
```

```
<div class="container">
```

```
<label class="hamburger"><span></span></label>
```

```
<div class="logo"><span class="wsite-logo">
```

```
<a href="/">
```

```
<span id="wsite-title">Breal</span>
```

```
</a>
```

```
</span></div>
```

```
<div class="nav desktop-nav"><ul class="wsite-menu-default">
```

```
<li id="active" class="wsite-menu-item-wrap ">
```

```
<a
```

```
href="/"
```

```
class="wsite-menu-item active"
```

```
>
```

Home

<li id="pg172601932864368916" class="wsite-menu-item-wrap ">

<a

href="/journal.html"

class="wsite-menu-item "

>

Journal

<li id="pg505244879255062737" class="wsite-menu-item-wrap ">

<a

href="/contact.html"

class="wsite-menu-item "

>

Contact

<li id="pg889357118968089600" class="wsite-menu-item-wrap ">

<a

```
                href="/store.html"

                class="wsite-menu-item "

            >

            STORE

        </a>

    </li>

</ul>

</div>

<div class="nav sidebar-nav"><div class="nav-wrap"><ul class="wsite-menu-default">

<li id="active" class="wsite-menu-item-wrap ">

<a href="/" class="wsite-menu-item active">Home</a>

</li>

<li id="pg172601932864368916" class="wsite-menu-item-wrap ">

<a href="/journal.html"      class="wsite-menu-item ">Journal</a>

</li>

<li id="pg505244879255062737" class="wsite-menu-item-wrap ">

<a href="/contact.html" class="wsite-menu-item ">Contact</a>

</li>

<li id="pg889357118968089600" class="wsite-menu-item-wrap ">

<a href="/store.html" class="wsite-menu-item ">STORE</a>

</li>
```


</div>

</div>

<div class="nav membership-cart"><span id="member-login"

class="wsite-custom-membership-wrapper"></div>

</div>

<div class="nav contact"></div>

</div>

</div>

<div class="main-wrap">

<div id="wsite-content" class="wsite-elements wsite-not-footer">

<div class="wsite-section-wrap">

<div class="wsite-section wsite-body-section wsite-background-3" >

<div class="wsite-section-content">

<div class="container">

<div class="wsite-section-elements">

<div class="commerce-elements-wrapper products__published"

data-page-id="470883442692997635" data-page-element-id="958692186878424604">

<div class="pagination__overlay"></div>

<div class="commerce-layout-wrapper">

```
<div class="product-grid product-grid-columns--3 product-grid-layout--above">

<div class="product-grid__item">

<a class="product-grid__item-overlay"

href="https://smartstore123.weebly.com/store/p2/HP_Pavilion_Gaming_Core_i7_8th_Gen_
html"></a>

<div class="product-grid__content">

<div class="product-grid__images">

<div class="product-grid-image image-1 product-grid-image__default
product-grid-image-aspect--1-1 js-default-img" style="background-image:
url(/uploads/1/2/4/8/124851663/s822957199410198092_p2_i1_w640.jpeg)">

<div class="product-grid__image-sale-banner-wrapper">

<p class="product-grid__image-sale-banner placeholder">

On Sale

</p>

</div>

<p class="product-grid__image-sale-banner visible">

On Sale

</p>

</div>

</div>

<div class="product-grid__info" style="text-align:left">

<h2 class="product-grid__title">
```

HP Pavilion Gaming Core i7 8th Gen

</h2>

<div class="product-grid__price paragraph">

\$975.00

\$770.00

</div>

<div class="product-grid__description paragraph">

8th Gen Intel Core i7-8750H Hexacore Processor(2.2 GHz base frequency, up to 4.1 GHz with Intel Turbo Boost Technology, 9 MB cache, 6 cores)8GB DDR4 RAM. Designed to run more efficiently and more reliably at faster speeds, DDR4 is the future of RAM128 GB M.2 SSD | 1 TB 5400 rpm SATANVIDIA GeForce GTX 1050 (4 GB GDDR5 dedicated) Discrete GraphicsThe notebook comes with preinstalled windows 10 home making it a ready to use device15.6 Inch diagonal Full HD SVA anti-glare WLED-backlit (1920 x 1080) 60 Hz Display3-cell, 52.2 Wh Li-ion HP Fast Charge Battery. Recharges your battery up to 50% within 30 minutes when the system is offDual HP speakers, custom-tuned in collaboration with experts at Bang & Olufsen, deliver rich, authentic sound

</div>

<div class="product-grid__button">

```
<a
href="https://smartstore123.weebly.com/store/p2/HP_Pavilion_Gaming_Core_i7_8th_Gen_.
html" class="wsite-button wsite-button-small wsite-button-highlight">

<span class="wsite-button-inner">Shop</span>

</a>

</div>

</div>

</div>

</div>

<div class="product-grid__item">

<a class="product-grid__item-overlay"
href="https://smartstore123.weebly.com/store/p3/Apple_MacBook_Pro_%2815-
inch_Retina%2C_Touch_Bar%2C_2.6GHz_6-
Core_Intel_Core_i7%2C_16GB_RAM%2C_512GB_SSD%29_-_Space_Gray.html"></a>

<div class="product-grid__content">

<div class="product-grid__images">

<div class="product-grid-image image-1 product-grid-image__default product-grid-image-
aspect--1-1 js-default-img" style="background-image:
url(/uploads/1/2/4/8/124851663/s822957199410198092_p3_i2_w640.jpeg)">

<div class="product-grid__image-sale-banner-wrapper">

<p class="product-grid__image-sale-banner placeholder">

On Sale

</p>
```


</div>

<p class="product-grid__image-sale-banner visible">

On Sale

</p>

</div>

</div>

<div class="product-grid__info" style="text-align:left">

<h2 class="product-grid__title">

Apple MacBook Pro

(15-inch Retina, Touch Bar, 2.6GHz 6-Core Intel Core i7, 16GB RAM, 512GB SSD) -
Space Gray

</h2>

<div class="product-grid__price paragraph">

\$2,649.00

\$2,549.00

</div>

<div class="product-grid__description paragraph">

```
<span class="product-grid-reset product-grid__description--shortened"><ul><li>6-core 8th-  
generation Intel Core i7 processor</li><li>Brilliant Retina display with True Tone  
technology</li><li>Touch Bar and Touch ID</li><li>Radeon Pro 555X or 560X graphics  
with 4GB of video memory</li><li>Ultrafast SSD</li><li>Intel UHD Graphics  
630</li><li>Four Thunderbolt 3 (USB-C) ports</li></ul></span>
```

```
</div>
```

```
<div class="product-grid__button">
```

```
<a href="https://smartstore123.weebly.com/store/p3/Apple_MacBook_Pro_%2815-  
inch_Retina%2C_Touch_Bar%2C_2.6GHz_6-  
Core_Intel_Core_i7%2C_16GB_RAM%2C_512GB_SSD%29_-_Space_Gray.html"  
class="wsite-button wsite-button-small wsite-button-highlight">
```

```
<span class="wsite-button-inner">Shop</span>
```

```
</a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="product-grid__item">
```

```
<a class="product-grid__item-overlay"  
href="https://smartstore123.weebly.com/store/p1/ASUS_VIVO_BOOKMAX_F541NA-  
GO6542.html"></a>
```

```
<div class="product-grid__content">
```

```
<div class="product-grid__images">
```

```
<div class="product-grid-image image-1 product-grid-image__default product-grid-image-  
aspect--1-1 js-default-img" style="background-image:  
url(/uploads/1/2/4/8/124851663/s822957199410198092_p1_i1_w640.jpeg)">
```

```
<div class="product-grid__image-sale-banner-wrapper">
```

```
<p class="product-grid__image-sale-banner placeholder">
```

On Sale

```
</p>
```

```
</div>
```

```
</script>
```

```
</body>
```

```
</html>
```

CHAPTER- 6

CHAPTER- 6

TESTING

6.1 Introduction to Testing:

The process of testing a software in a well-planned and systematic way is known as software testing lifecycle (STLC).

Different organizations have different phases in STLC however generic Software Test Life Cycle (STLC) for waterfall development model consists of the following phases.

1. Requirements Analysis
2. Test Planning
3. Test Analysis
4. Test Design
5. Test Construction and Verification
6. Test Execution and Bug Reporting
7. Final Testing and Implementation
8. Post Implementation

1. Requirements Analysis:

In this phase testers analyse the customer requirements and work with developers during the design phase to see which requirements are testable and how they are going to test those requirements.

It is very important to start testing activities from the requirements phase itself because the cost of fixing defect is very less if it is found in requirements phase rather than in future phases.

2. Test Planning:

In this phase all the planning about testing is done like what needs to be tested, how the testing will be done, test strategy to be followed, what will be the test environment, what test methodologies will be followed, hardware and software availability, resources, risks etc. A high level test plan document is created which includes all the planning inputs mentioned above and circulated to the stakeholders.

Usually IEEE 829 test plan template is used for test planning.

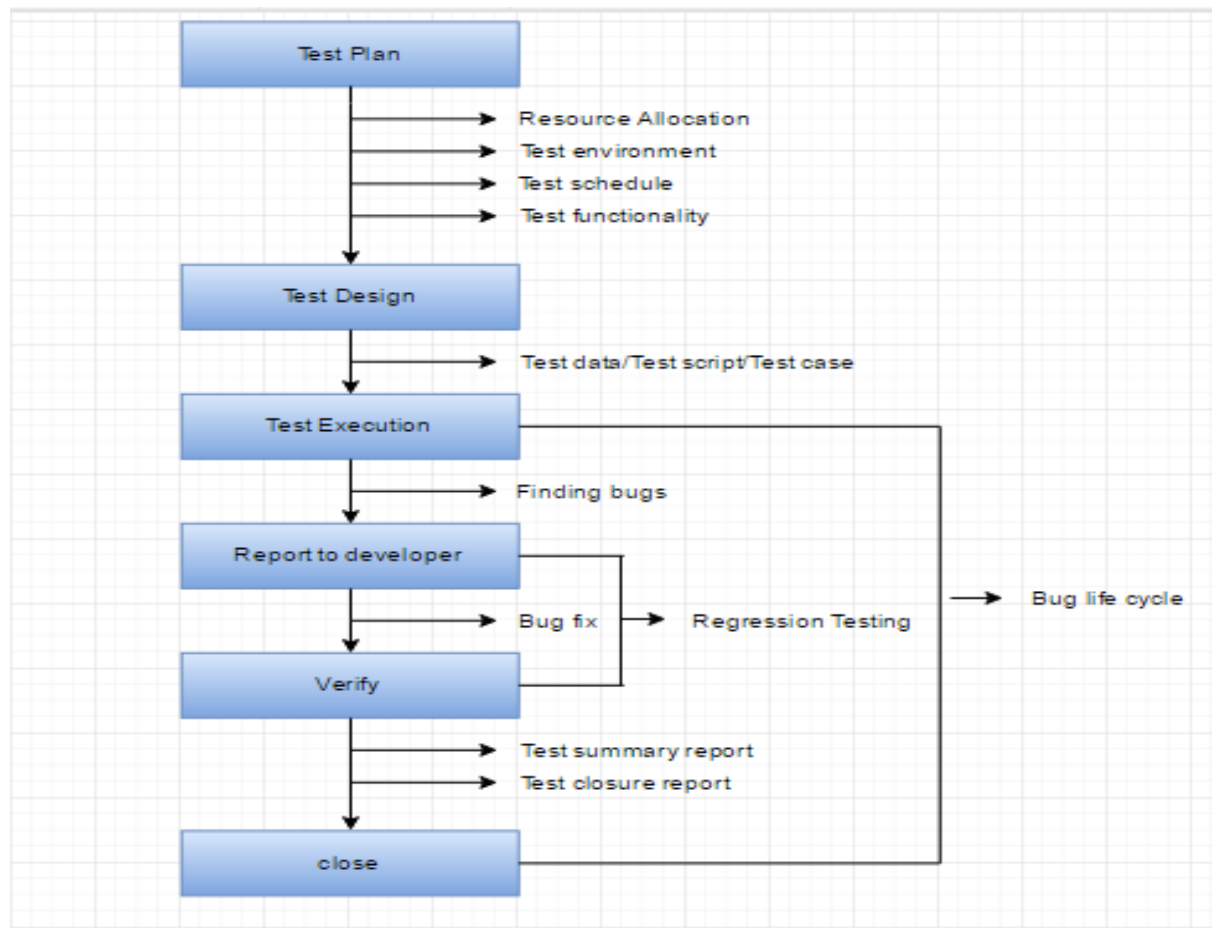


Figure 6.1: Test Planning

3. Test Analysis:

After test planning phase is over test analysis phase starts, in this phase we need to dig deeper into project and figure out what testing needs to be carried out in each SDLC phase.

Automation activities are also decided in this phase, if automation needs to be done for software product, how will the automation be done, how much time will it take to automate and which features need to be automated. Non-functional testing areas (Stress and performance testing) are also analysed and defined in this phase.

4. Test Design:

In this phase various black-box and white-box test design techniques are used to design the test cases for testing, testers start writing test cases by following those design techniques, if automation testing needs to be done then automation scripts also needs to written in this phase.

5. Test Construction and Verification:

In this phase testers prepare more test cases by keeping in mind the positive and negative scenarios, end user scenarios etc. All the test cases and automation scripts need to be complete in this phase and got reviewed by the stakeholders. The test plan document should also be finalized and verified by reviewers.

6. Test Execution and Bug Reporting:

Once the unit testing is done by the developers and test team gets the test build, the test cases are executed and defects are reported in bug tracking tool, after the test execution is complete and all the defects are reported. Test execution reports are created and circulated to project stakeholders.

After developers fix the bugs raised by testers they give another build with fixes to testers, testers do re-testing and regression testing to ensure that the defect has been fixed and not affected any other areas of software.

Testing is an iterative process i.e. If defect is found and fixed, testing needs to be done after every defect fix.

After tester assures that defects have been fixed and no more critical defects remain in software the build is given for final testing.

7. Final Testing and Implementation:

In this phase the final testing is done for the software, non-functional testing like stress, load and performance testing are performed in this phase. The software is also verified in the production kind of environment. Final test execution reports and documents are prepared in this phase.

8. Post Implementation:

In this phase the test environment is cleaned up and restored to default state, the process review meetings are done and lessons learnt are documented. A document is prepared to cope up similar problems in future releases.

Phase	Activities	Outcome
Planning	Create high level test plan	Test plan, Refined Specification
Analysis	Create detailed test plan, Functional Validation Matrix, test cases	Revised Test Plan, Functional validation matrix, test cases
Design	test cases are revised; select which test cases to automate	revised test cases, test data sets, sets, risk assessment sheet
Construction	scripting of test cases to automate,	test procedures/Scripts, Drivers, test results, Bug Reports.
Testing cycles	complete testing cycles	Test results, Bug Reports

Final testing	execute remaining stress and performance tests, complete documentation	Test results and different metrics on test efforts
Post implementation	Evaluate testing processes	Plan for improvement of testing process

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not.

Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

According to ANSI/IEEE 1059 standard, Testing can be defined as - A process of analysing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.

Who does Testing?

It depends on the process and the associated stakeholders of the project(s). In the IT industry, large companies have a team with responsibilities to evaluate the developed software in context of the given requirements. Moreover, developers also conduct testing which is called **Unit Testing**. In most cases, the following professionals are involved in testing a system within their respective capacities:

- Software Tester
- Software Developer
- Project Lead/Manager
- End User

Different companies have different designations for people who test the software on the basis of their experience and knowledge such as Software Tester, Software Quality Assurance Engineer, QA Analyst, etc.

It is not possible to test the software at any time during its cycle. The next two sections state when testing should be started and when to end it during the SDLC.

When to Start Testing?

An early start to testing reduces the cost and time to rework and produce error-free software that is delivered to the client. However, in Software Development Life Cycle (SDLC), testing can be started from the Requirements Gathering phase and continued till the deployment of the software. It also depends on the development model that is being used. For example, in the Waterfall model, formal testing is conducted in the testing phase; but in the incremental model, testing is performed at the end of every increment/iteration and the whole application is tested at the end.

Testing is done in different forms at every phase of SDLC:

- During the requirement gathering phase, the analysis and verification of requirements are also considered as testing.
- Reviewing the design in the design phase with the intent to improve the design is also considered as testing.
- Testing performed by a developer on completion of the code is also categorized as testing.

When to Stop Testing?

It is difficult to determine when to stop testing, as testing is a never-ending process and no one can claim that software is 100% tested. The following aspects are to be considered for stopping the testing process:

- Testing Deadlines
- Completion of test case execution
- Completion of functional and code coverage to a certain point
- Bug rate falls below a certain level and no high-priority bugs are identified
- Management decision

Quality Assurance:

QA includes activities that ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements.

Types of Testing:

This section describes the different types of testing that may be used to test software during SDLC.

Manual Testing:

Manual testing includes testing a software manually, i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behaviour or bug. There are different stages for manual testing such as unit testing, integration testing, system testing, and user acceptance testing.

Testers use test plans, test cases, or test scenarios to test a software to ensure the completeness of testing. Manual testing also includes exploratory testing, as testers explore the software to identify errors in it.

Automation Testing:

Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses another software to test the product. This process involves automation of a manual process. Automation Testing is used to re-run the test scenarios that were performed manually, quickly, and repeatedly.

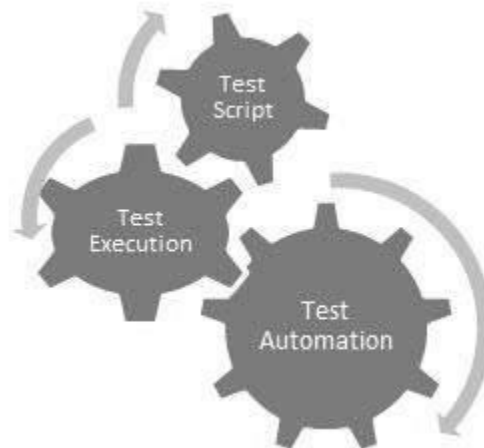


Figure 6.2: Diagram for Automation Testing

Apart from regression testing, automation testing is also used to test the application from load, performance, and stress point of view. It increases the test coverage, improves accuracy, and saves time and money in comparison to manual testing.

What is Automate?

It is not possible to automate everything in a software. The areas at which a user can make transactions such as the login form or registration forms, any area where large number of users can access the software simultaneously should be automated.

Furthermore, all GUI items, connections with databases, field validations, etc. can be efficiently tested by automating the manual process.

When to Automate?

Test Automation should be used by considering the following aspects of a software:

- Large and critical projects
- Projects that require testing the same areas frequently
- Requirements not changing frequently
- Accessing the application for load and performance with many virtual users

- Stable software with respect to manual testing
- Availability of time

How to Automate?

Automation is done by using a supportive computer language like VB scripting and an automated software application. There are many tools available that can be used to write automation scripts. Before mentioning the tools, let us identify the process that can be used to automate the testing process:

- Identifying areas within a software for automation
- Selection of appropriate tool for test automation
- Writing test scripts
- Development of test suits
- Execution of scripts
- Create result reports
- Identify any potential bug or performance issues

Software Testing Tools:

The following tools can be used for automation testing:

- Selenium
- IBM Rational Functional Tester
- Silk Test
- Test Complete
- Testing Anywhere
- Win Runner
- Load Runner
- Visual Studio Test Professional
- WATIR
- HP Quick Test Professional

There are different methods that can be used for software testing.

Black-Box Testing:

The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

White-Box Testing:

White-box testing is the detailed investigation of internal logic and structure of the code. White-box testing is also called **glass testing** or **open-box testing**. In order to perform **white-box** testing on an application, a tester needs to know the internal workings of the code.

The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

Grey-Box Testing:

Grey-box testing is a technique to test the application with having a limited knowledge of the internal workings of an application. In software testing, the phrase the more you know, the better carries a lot of weight while testing an application.

Mastering the domain of a system always gives the tester an edge over someone with limited domain knowledge. Unlike black-box testing, where the tester only tests the application's user interface; in grey-box testing, the tester has access to design documents and the database. Having this knowledge, a tester can prepare better test data and test scenarios while making a test plan.

Levels of testing include different methodologies that can be used while conducting software testing. The main levels of software testing are:

- Functional Testing
- Non-functional Testing

Functional Testing:

This is a type of black-box testing that is based on the specifications of the software that is to be tested. The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for. Functional testing of a software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

We have 7 different types of KPI's. They are as follows:

1. Bounce Rate:











Bounce Rate by Page  		
Page	Bounce Rate	Bounces
/?fbclid=IwAR26QPNT6htv9ddoHaenj8UftR0B-IO7TsrA8akl21oDcDzIO1rfwG5KeCl 	100.00%	1
/blog/the-petrifying-we-ll 	100.00%	2
/blog/underwater-park 	100.00%	1
/who-we-are.html 	100.00%	1
/blog/never-ending-lightning-storms 	50.00%	1
/blog/mystery-of-the-dead-evils-kettle 	33.33%	1
/ 	29.63%	72
/blog/boiling-river 	20.00%	1

Figure 6.3: Bounce Rate per page

This KPI is used to describe the bounce rate of pages from which visitors are bouncing out most.

The dimensions used in this KPI is Page and metric is Bounces and Bounce Rate.

Page: A Page refers to the one of the page of your website.

Bounces: It is the visits in which a person leaves your website from the landing page without browsing any further.

Bounce Rate: It is the percentage of visits in which a person leaves your website from the landing page without browsing any further.

2. Number of visits per page:

What pages do your users visit?

Page	Page Views	Page Value
/	12	\$0.00
/shopnow.html	5	\$0.00
/contact.html	1	\$0.00
/journal	1	\$0.00

Last 7 days ▾ [PAGES REPORT >](#)

Figure 6.4: Number of visits per page

This KPI is used to describe the number of times a page is being visited by users mostly.

The dimensions used in this KPI is Page and metric is Page Views and Page Value.

Page Views: It is the number of visits of all the users to one particular page.

Page Value: It is the vale assigned to any page by the admin for business purpose.

3. Number of sessions:

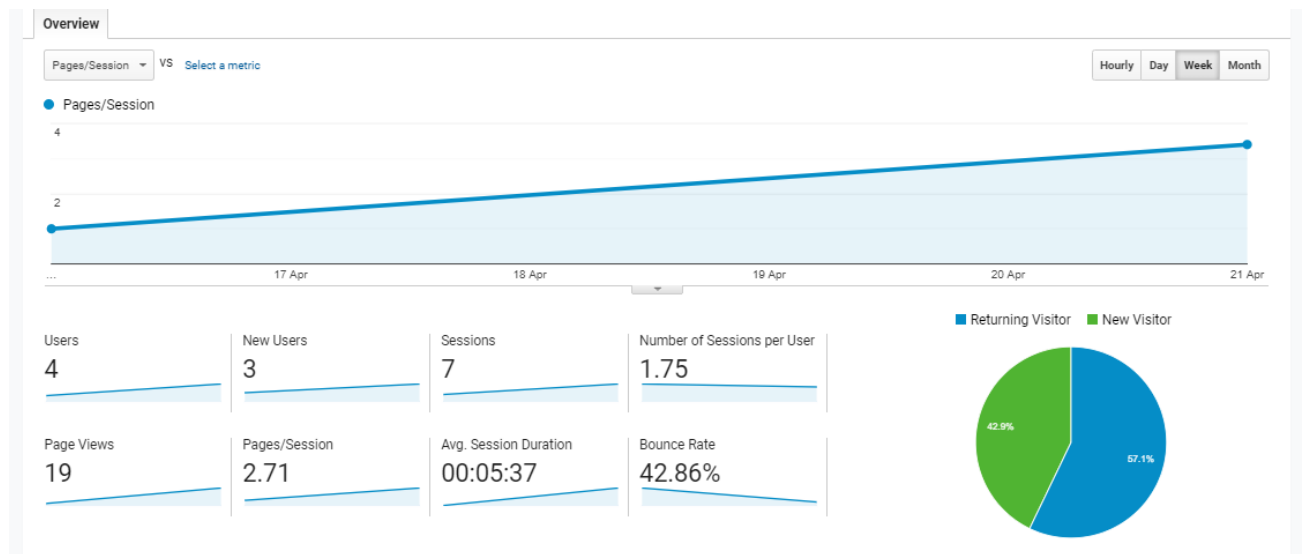


Figure 6.5: Number of sessions

This KPI is used to describe the total number of pages being visited by users.

The dimensions used in this KPI is Page and metric is Page Views.

Page: A Page refers to the one of the page of your website.

Page Views: It is the number of visits of all the users to one particular page.

4. Sessions by device:

This KPI is used to describe the number of times a page is being visited by users based on their hardware specification i.e., through website or through mobile.

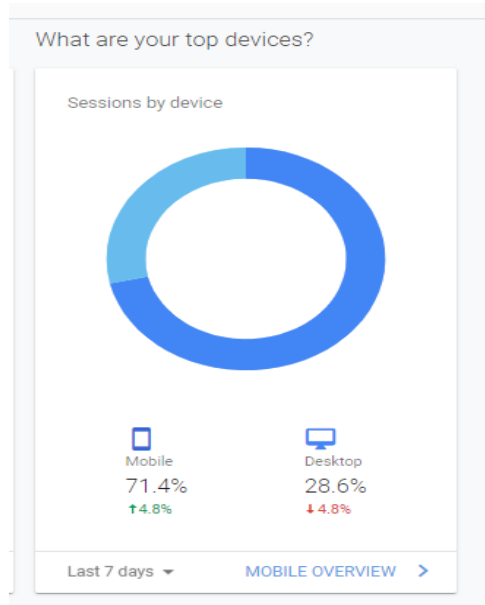


Figure 6.6: Sessions by device

The dimensions used in this KPI is Page and metric is Page Views.

Page: A Page refers to the one of the page of your website.

Page Views: It is the number of visits of all the users to one particular page.

5. Sessions by country:

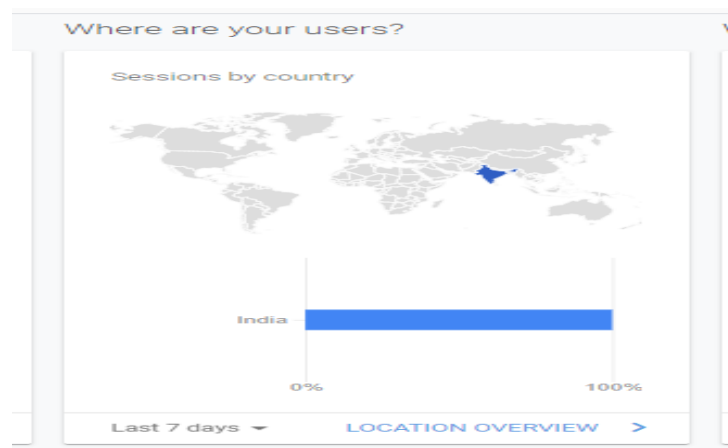


Figure 6.7: Sessions by country

This KPI is used to describe the number of times a page is being visited by users based on their location i.e., the country or city they are browsing the website.

The dimensions used in this KPI is Region and metric is Users.

Page: A Page refers to the one of the page of your website.

Page Views: It is the number of visits of all the users to one particular page.

6. Users by time of day:

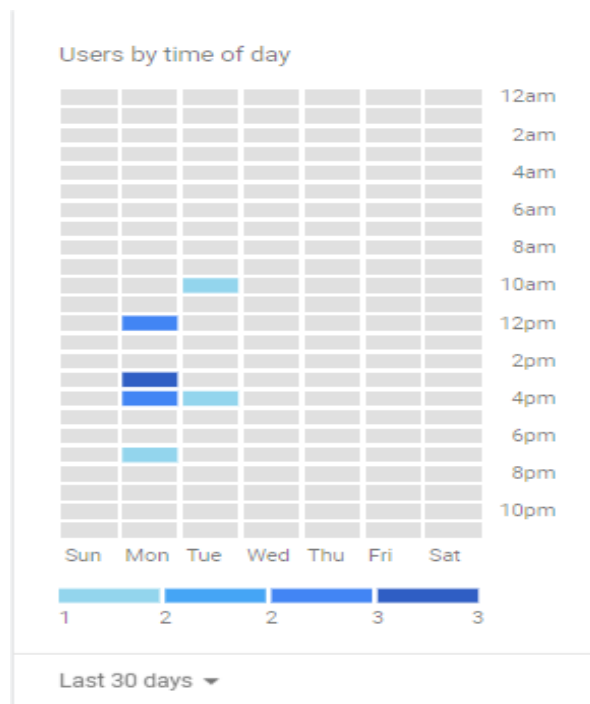


Figure 6.8: Users by time of the day

This KPI is used to describe the number of times a page is being visited by users based on the time they are browsing the website.

The dimensions used in this KPI is Page and metric is Page Views and Time Stamp.

Page: A Page refers to the one of the page of your website.

Page Views: It is the number of visits of all the users to one particular page.

Time Stamp: It is the current time of the system that is retrieved for statistical purpose.

7. Number of users accessing the website in real time:

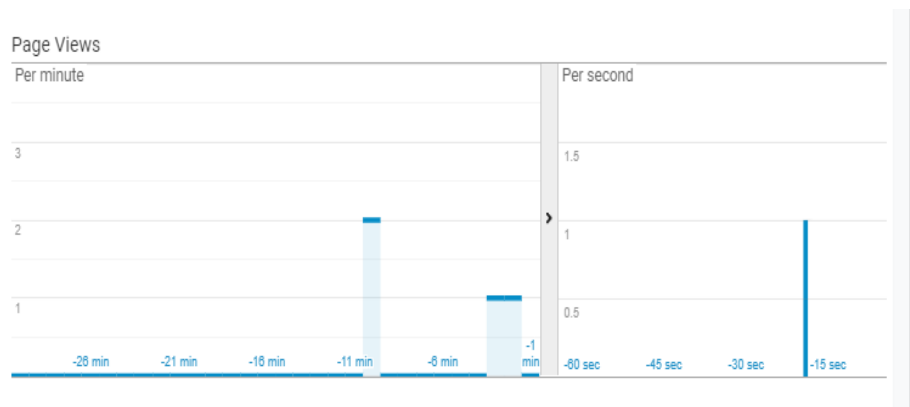


Figure 6.9: Number of users accessing the website in real time

This KPI is used to describe the number of times a page is being visited by users based on the current time and generating graphs on minutes and seconds basis.

The dimensions used in this KPI is Page and metric is Page Views and Time Stamp.

Page: A Page refers to the one of the page of your website.

Page Views: It is the number of visits of all the users to one particular page.

Time Stamp: It is the current time of the system that is retrieved for statistical purpose.

12. Sessions:

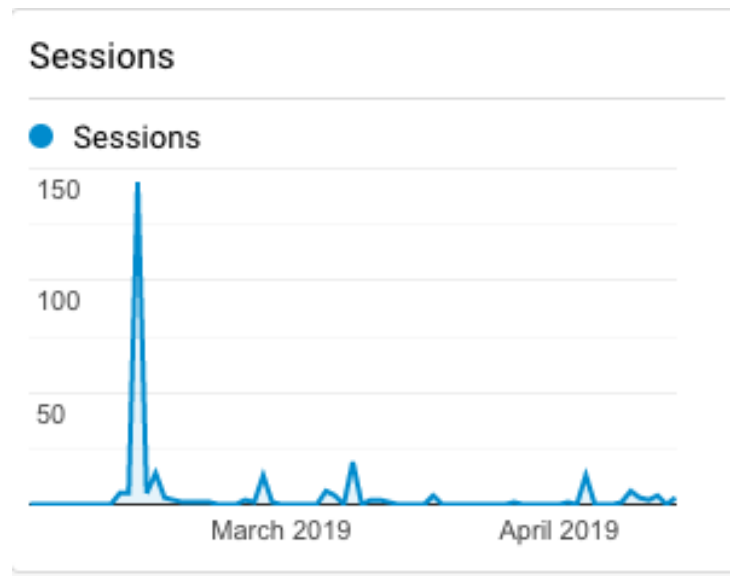


Figure 6.10: Number of sessions based on date

This KPI is used to show the Total number of Sessions. The metric used in this KPI is Sessions.

Sessions: A session is defined as a group of interactions one user takes within a given time frame on your website. Google Analytics defaults that time frame to 30 minutes.

6.2 Test Cases:

A **test case** is a document, which has a set of test data, preconditions, expected results and post conditions, developed for a particular test scenario in order to verify compliance against a specific requirement.

Below are the standard fields of sample test case template:

A **test case** is a document, which has a set of test data, preconditions, expected results and post conditions, developed for a particular test scenario in order to verify compliance against a specific requirement.

Below are the standard fields of sample test case template.

Module Name	: Mobile E- Commerce using Google Analytics.
Test Designed by	: Annaparthi Vyshnavi
Test Designed Date	: 20 th January
Test Executed by	: Heena Fatima
Test Execution Date	: 25 th January
Test Title/Name	: <u>smartstore123.weebly.com</u>
Test Summary/Description	: To track the data and to maintain and run the Website successfully.
Expected Result	: The website should be accessible and Google analytics should track the data and generate reports.
Actual result	: Yes, the website is accessible to everyone and Google analytics server is tracking the data.
Status (Pass/Fail)	: The test conducted is successfully passed.

CHAPTER- 7

CHAPTER- 7

SCREENSHOTS

7.1 Home Page:

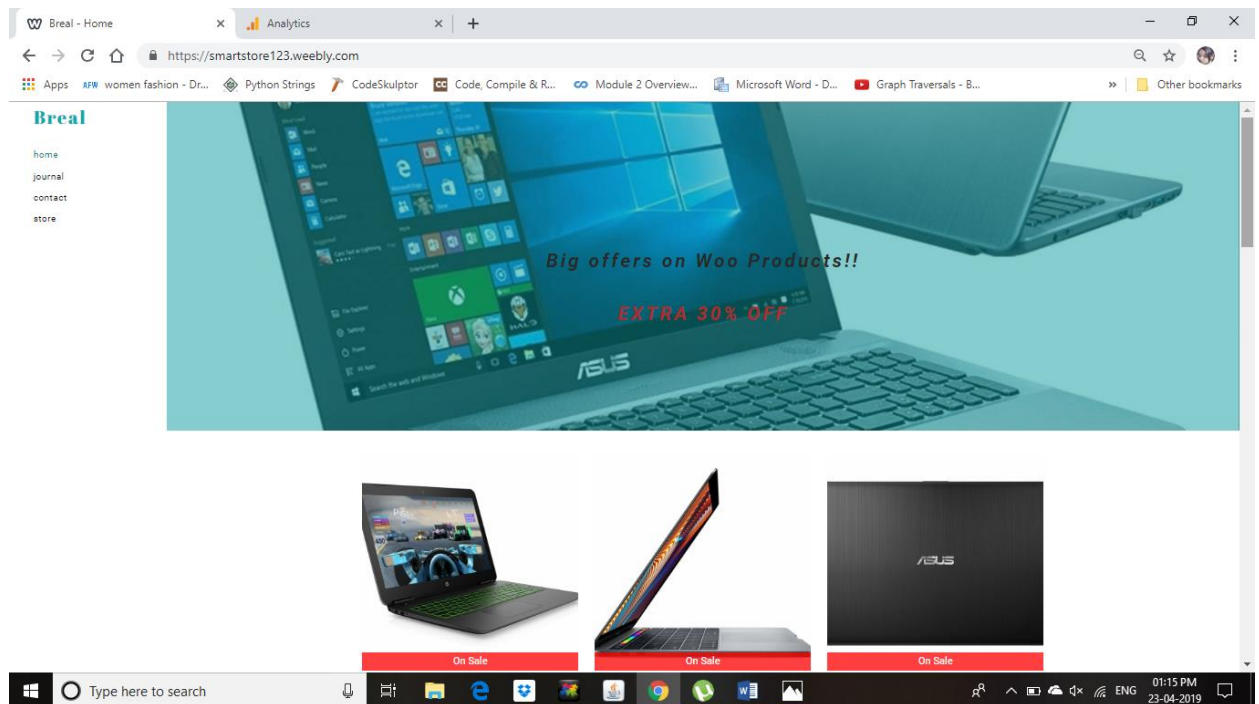


Figure 7.1: Home Page of the website

7.2 Journal Page:

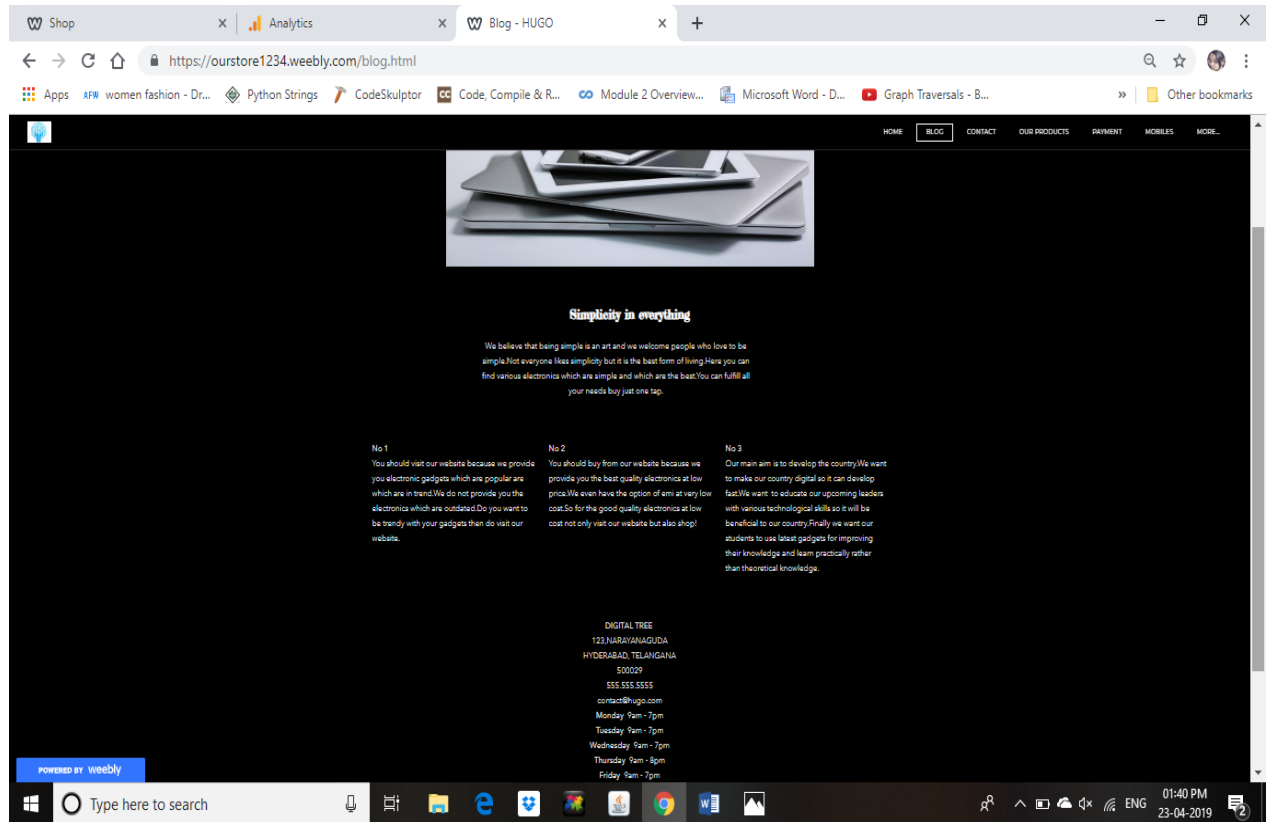


Figure 7.2: Journal Page to describe about the website

7.3 Contact Page:

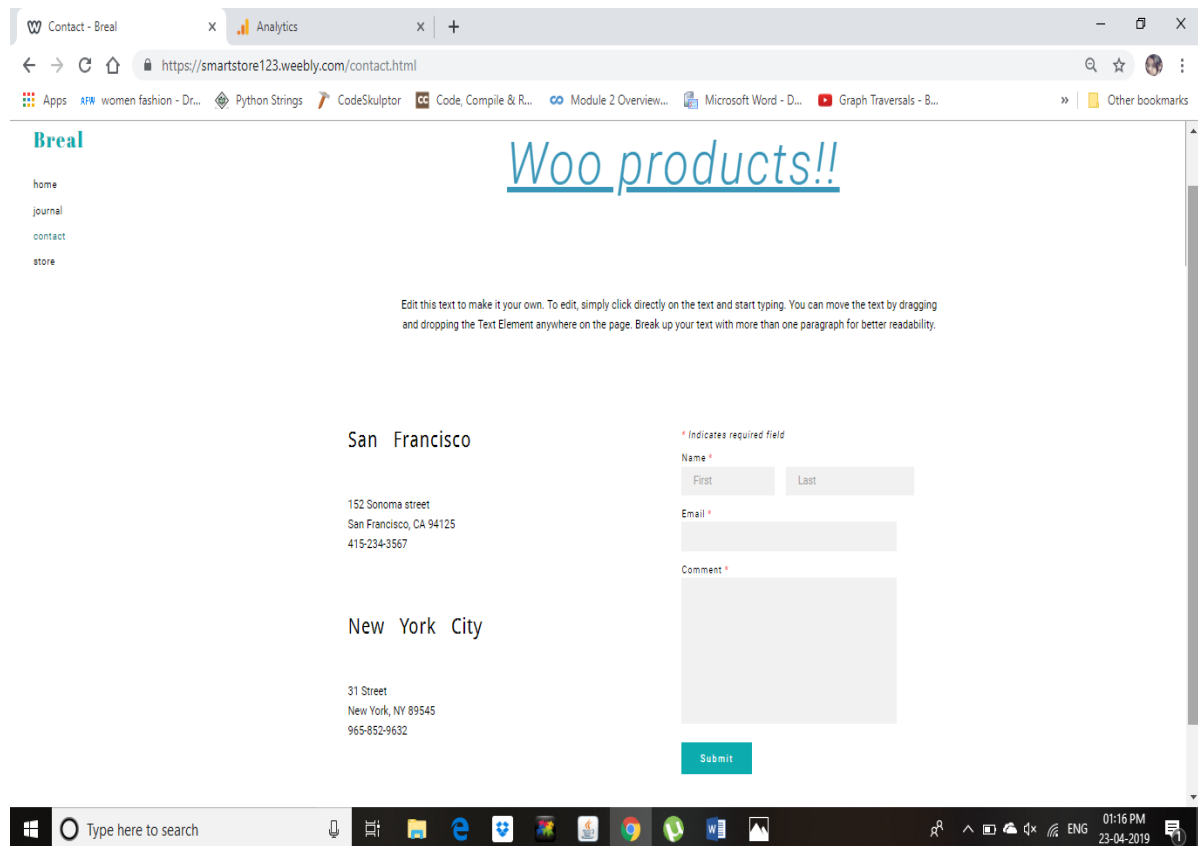


Figure 7.3: Contact Page

7.4 Store Page:

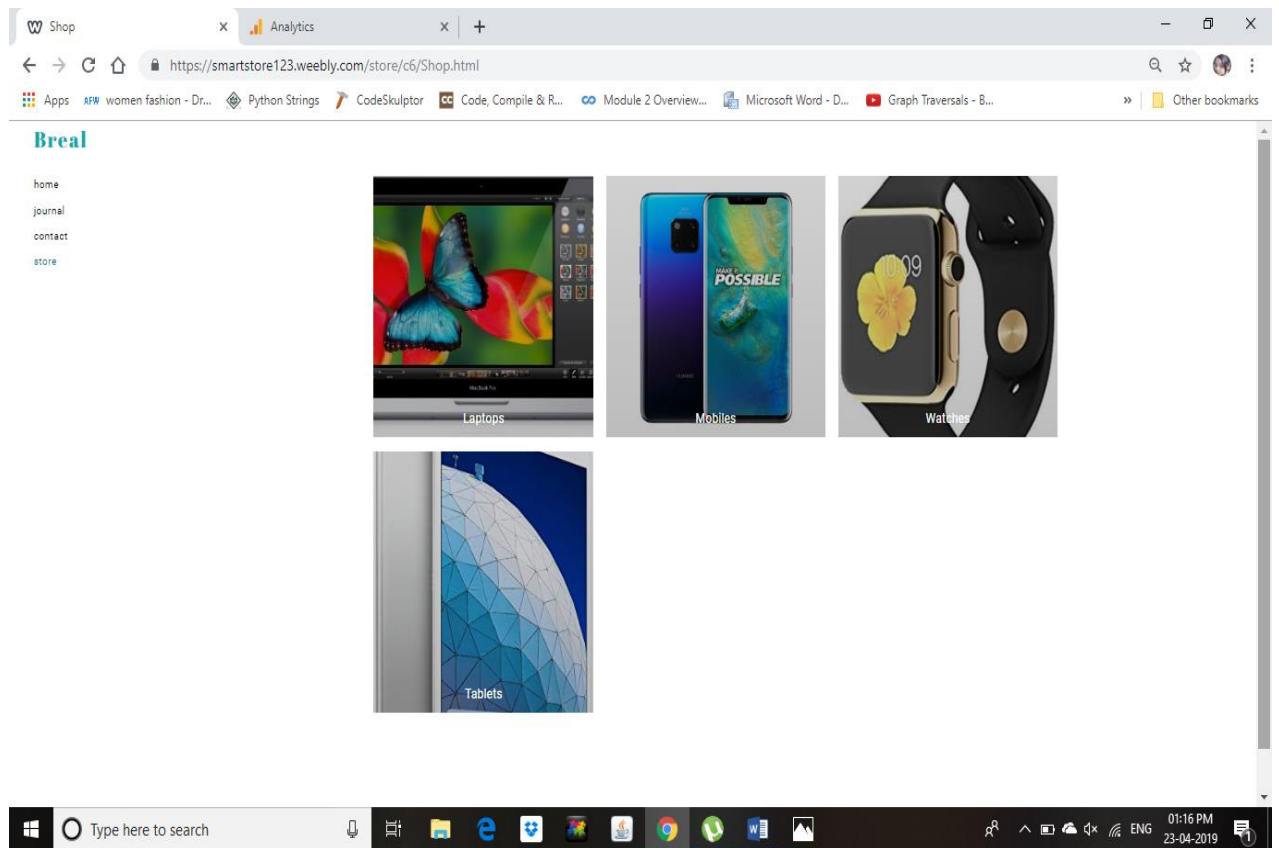


Figure 7.4: Store Page

Google Dashboard:

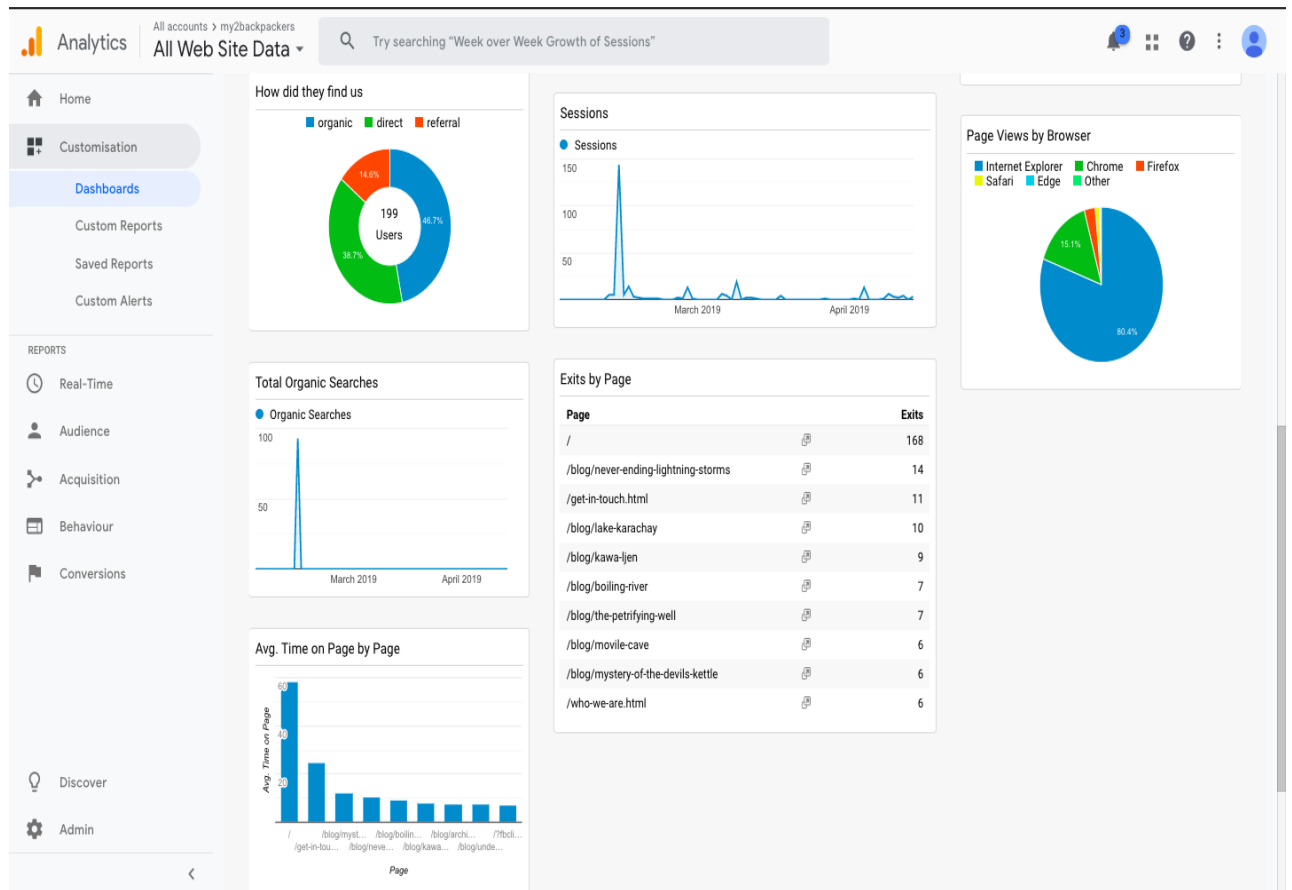


Figure 7.5: Google Dashboard

Reports: Audience Report:

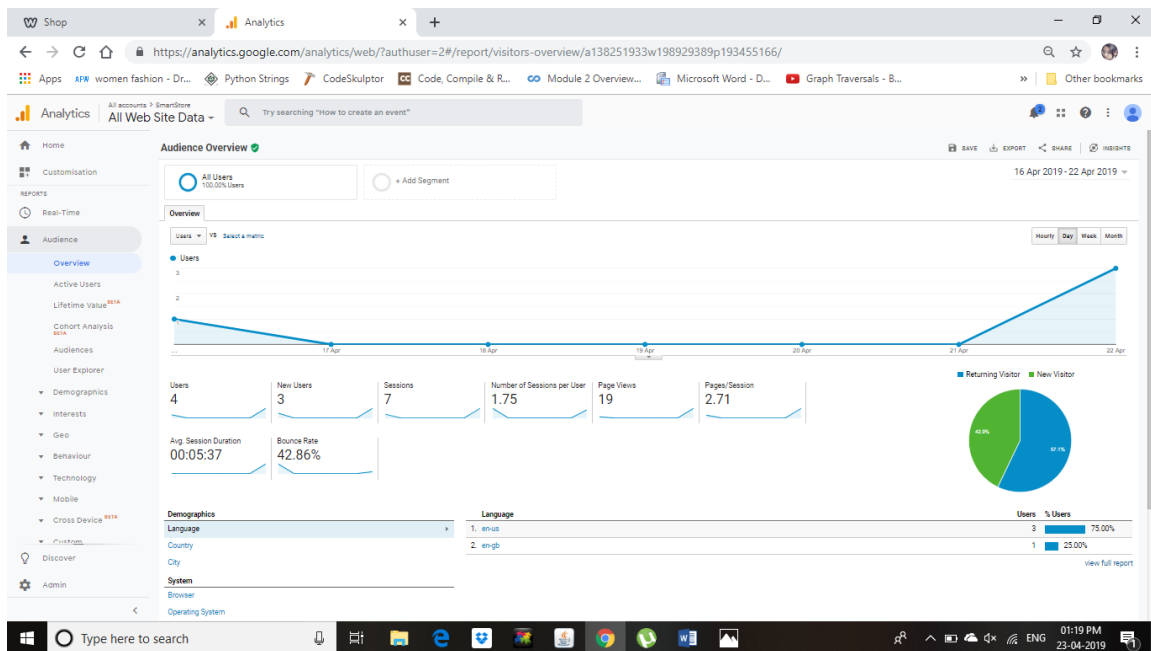


Figure 7.6: Audience Report

Acquisition Report:

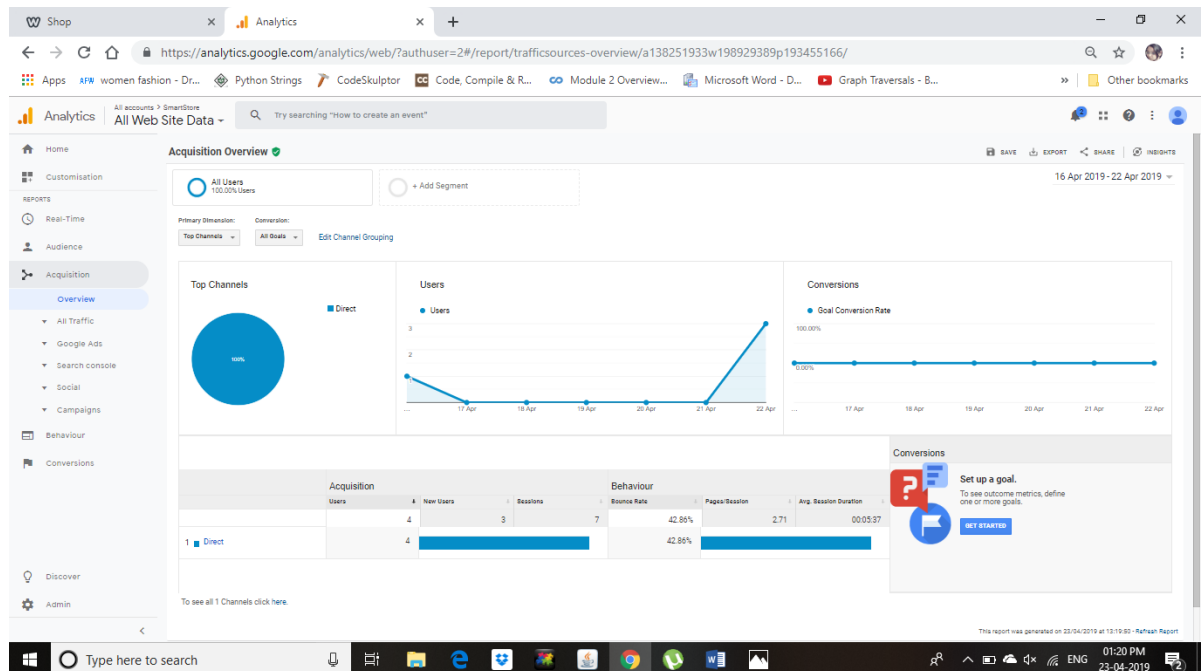


Figure 7.7: Acquisition Report

Mobile E- Commerce using Google Analytics

Behaviour Report:

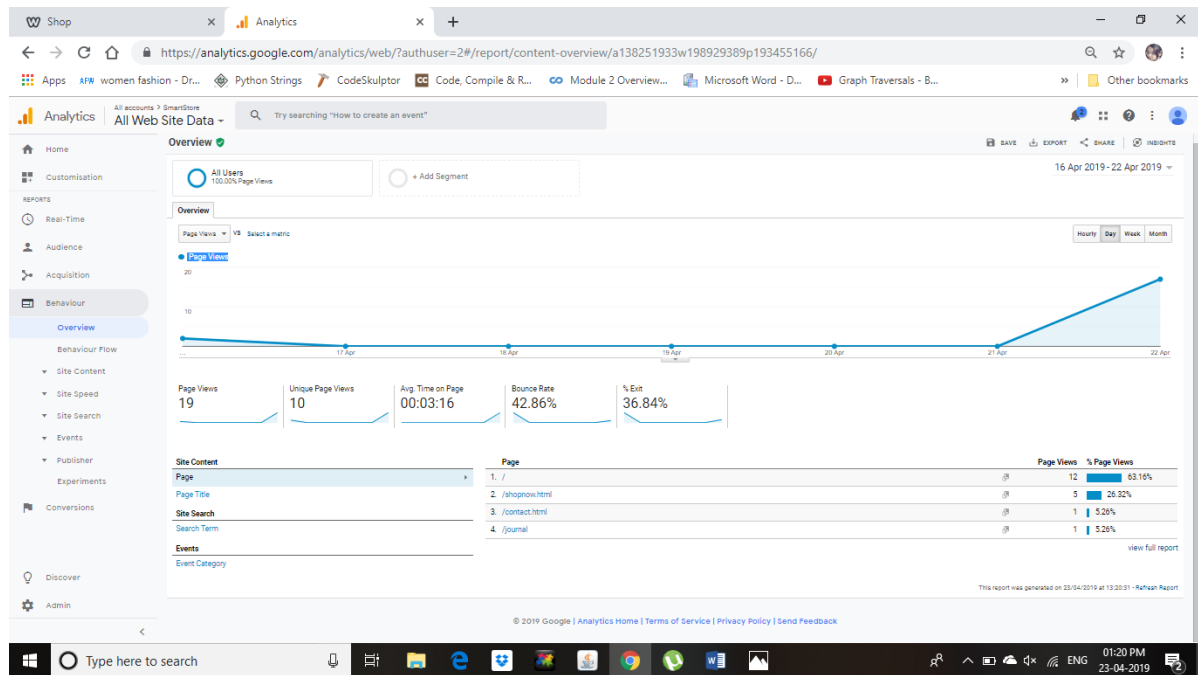


Figure 7.8: Behaviour Report

Real Time Report:

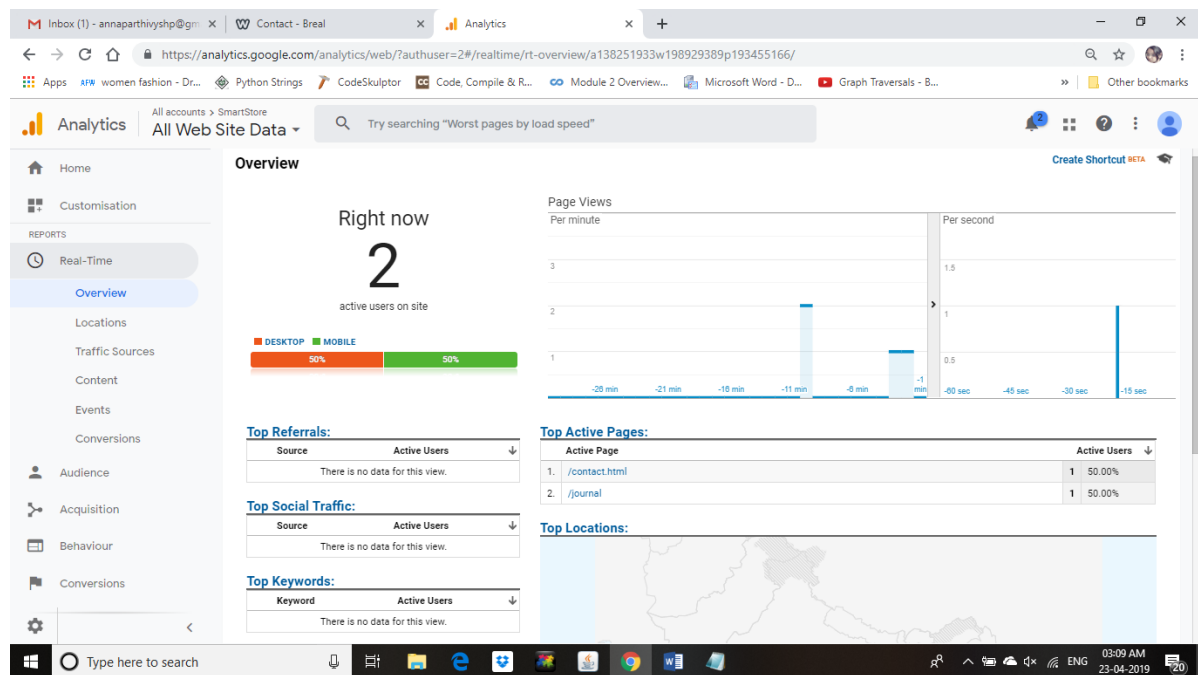


Figure 7.9: Real Time Report

CONCLUSION

Whenever we are ready host a Website, one thing that always bother you, “*How do you reach the audience?*”. To answer this question, we need to track our website, generate reports based on user location, accessing devices, type of browser it supports, type of device, etc. by considering different factors that affect the performance of website. Manually we cannot track the website and generating the reports. In order to track the website, we are using a freemium web analytics tool ‘GOOGLE ANALYTICS’. The events performed on our website will be tracked and stored in GA.

By using the recorded information, we can analyse the data and present it on Dashboard. We can generate the reports as Tables, pie-charts, Maps, Bar graph. Line graph, etc... These reports will help you to find the areas we need to improve and do the proper changes to reach out all kinds of audience. We eventually will be developing software to support different technologies.

So, we can say Web Analytics is a useful tool to track the website, and to store the tracked information, and to generate reports. To bring awareness to Host of the website, to improve the lagged areas and can reach out the audience well.

REFERENCES

1. <https://www.w3schools.com>
2. <https://www.w3.org/Style/CSS/Overview.en.html>
3. <https://www.javascript.com/>
4. <https://www.weebly.com/home>
5. <https://analytics.google.com/analytics/web/#/>
6. <https://www.google.com/analytics/tag-manager/use-policy/>

BIBLIOGRAPHY

- [1] Muret, Paul (2016-03-15). "Introducing the Google Analytics 360 suite". Google. Retrieved 2018-10-07.
- [2] Press, The Associated (2005-03-29). "Google Acquires Urchin Software". The New York Times. ISSN 0362-4331. Retrieved 2019-04-04.
- [3] How do Goals work in Analytics and Adwords?, Cerebro Marketing, archived from the original on March 6, 2016, retrieved February 17, 2016.
- [4] "Google Analytics for Mobile Apps | Analytics Implementation Guides and Solutions | Google Developers". Google Developers. Retrieved 2017-08-25.
- [5] Greenberg, Andy (December 11, 2008). "The Virus Filters". Forbes.
- [6] Muret, Paul (January 20, 2012). "The End of an Era for Urchin Software". Google Analytics. Retrieved 2012-04-07.
- [7] "Tracking Code: The _gat Global Object". Google. January 24, 2012. Retrieved 2012-06-27.
- [8] "Google Tag Manager: A Step-By-Step Guide | Analytics & Optimization". online-behavior.com. Retrieved 2016-03-17.
- [9] "Google Analytics Launches Asynchronous Tracking - The official Google Code blog". 1 December 2009.