Z’elle

TechNoStress

A Project by Miss. Ginelle D'souza

Table Of Content

CHAPTER 1

INTRODUCTION

1.1. Synopsis of the project

1.1.1. About the project

In this fast pace world, we have the least time to research over the ever-increasing new technologies and gadgets. The purpose of this project is to provide assistance to electronic buyer specifically in the area of mobile devices, laptops and tablets. Z'elle proves to be a one stop solution that allows the user to choose a desired produced from a wide range of variety. Z'elle brings a range of optimal technologies that have been purchased, used, and loved by many. The market is flooded with an ocean of products. These products are offered by big companies such as Apple, Samsung, OPPO, Vivo, OnePlus, Dell, HP, etc. Due to the increasing competition these companies tend to launch more and more products, thus drowning the end users in their ocean of technologies. Z'elle poses are a floater for the end user by making exploring easier through machine learning algorithms and recommendation systems. These algorithms produce a set of recommended technologies that would be most feasible to the user.

1.2 Objective & Scope of Project

Objective

The main goal of this website is to provide ease to the end user while making a choice to purchase electronic products such as mobile phones, personal computers and tablet phones.

Scope

Z'elle makes use of "machine learning algorithms" to study and identify the behavior of the end user. The project provides an additional feature that caters to the users and provides a set of recommended technologies. This project makes use of web scraping the provide current trending technical news.

- **Content Based Recommendation System**

  A Content-Based Recommendation is used to suggest relevant data based on only and only the features of the entity searched.

- **Machine Learning**

  Machine learning is the study of algorithms that helps to improve the experience of the user and provide accurate results by the use of data.

- **Web Scrapping**

  Web scraping is the process of using scripts or bots to extract content and data from the internet.

1.3 Problem Definition

Existing Technology

Several electronic shopping websites like Flipkart, Amazon, Alibaba, etc., provide enormous digital data. These websites provide a large variety of information for electronics. Users can use this information and filter the data according their preferences. But the problem arises when a user isn't sure what they want or even whether a particular product is necessary for them. There may be a situation when the user isn't very tech savvy and doesn't understand the information provided to them, in such cases it is very important to have a person guiding them so that the user is able to make the best choice of purchase.

Proposed Technology

Z'elle brings to you a wide range of optimal technologies that have been purchased, used, and loved by many. These enlisted gadgets are pocket-friendly and user dedicated, based on your requirements. This project proposes a "Gadget Predictor" that works like magic. Z'elles own "Gadget Predictor" makes use of Artificial Intelligence to provide a list of potential gadgets that would be optimal for the end users use in no time. What you have to do? Nothing. According to the needs of the user, Z'elle will provide the user a variety of solution so that truly Z'elle can be a "**One Stop Technology**"

1.4 Theoretical Background

The main language used for the development of Z'elle is Python 3.8.5. Flask web framework supports to host this website on the localhost. Z'elle stores its information extracted the users in Amazon Relational Database Service (Amazon RDS) – PostgreSQL. The user interface has been built on the strong base of HTML 5 and beautified by CSS, as well as JavaScript. The prediction made by the algorithm is through a pickle Random Forest machine learning technique. In order to present trending/current news web scrapping is done through BeautifulSoup.

1.4.1 Overview of Front End

HTML

The HyperText Markup Language or HTML is a standard markup language used for documents to be designed and displayed by a web browser.

CSS

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.

JavaScript

JavaScript is a programming language commonly used in web development. It

was originally developed by Netscape as a means to add dynamic and interactive elements to websites

Google Data Studio

Google Data Studio is an online tool for converting data into customizable informative reports and dashboards introduced by Google.

1.4.2 Overview of Back End

Python

Python is a general-purpose, versatile, and powerful programming language. It's a great first language because it's concise and easy to read.

Flask

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications.

BeautifulSoup

Beautiful Soup is a Python package for parsing HTML and XML documents. It creates a parse tree to extract data from HTML, which is useful for web scraping

Amazon RDS – PostgresSQL

Amazon Relational Database Service is a distributed relational database service by Amazon Web Services. It is a web service running "in the cloud" designed to simplify the setup, operation, and scaling of a relational database for use in applications.

Random Forest - Machine Learning

Random forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because of its simplicity and diversity

CHAPTER 2

SYSTEM ANALYSIS

2.1 Feasibility Study

2.1.1 Technical Feasibility

The feasibility study is basically the test of the proposed system in the light of its workability, meeting user's requirements, effective use of resources and of course, the cost effectiveness. These are categorized as technical, operational, economic, schedule and social feasibility. The main goal of feasibility study is not to solve the problem but to achieve the scope. In the process of feasibility study, the cost and benefits are estimated with greater accuracy to find the Return on Investment (ROI). This also defines the resources needed to complete the detailed investigation. The result is a feasibility report submitted to the management. This may be accepted or accepted with modifications or rejected.

2.1.2 Economic Feasibility

Economic feasibility is a kind of cost-benefit analysis of the examined project, which assesses whether it is possible to implement it. This term means the assessment and analysis of a project's potential to support the decision-making process by objectively and rationally identifying its strengths, weaknesses, opportunities and risks associated with it, the resources that will be needed to implement the project, and an assessment of its chances of success. It consists of market analysis, economic analysis, technical and strategic analysis.

2.1.3 Operational Feasibility

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. Operational feasibility reviews the willingness of the organization to support the proposed system. This is probably the most difficult of the feasibilities to gauge. In order to determine this feasibility, it is important to understand the management commitment to the proposed project. If the request was initiated by management, it is likely that there is management support and the system will be accepted and used. However, it is also important that the employee base will be accepting of the change.

2.2 Cost Benefit Analysis

A cost benefit analysis (also known as a benefit cost analysis) is a process by which organizations can analyze decisions, systems or projects, or determine a value for intangibles. The model is built by identifying the benefits of an action as well as the associated costs, and subtracting the costs from benefits. When completed, a cost benefit analysis will yield concrete results that can be used to develop reasonable conclusions around the feasibility and/or advisability of a decision or situation.

As we know that the system development costs are usually one-time costs that will not recur after the project has been completed.

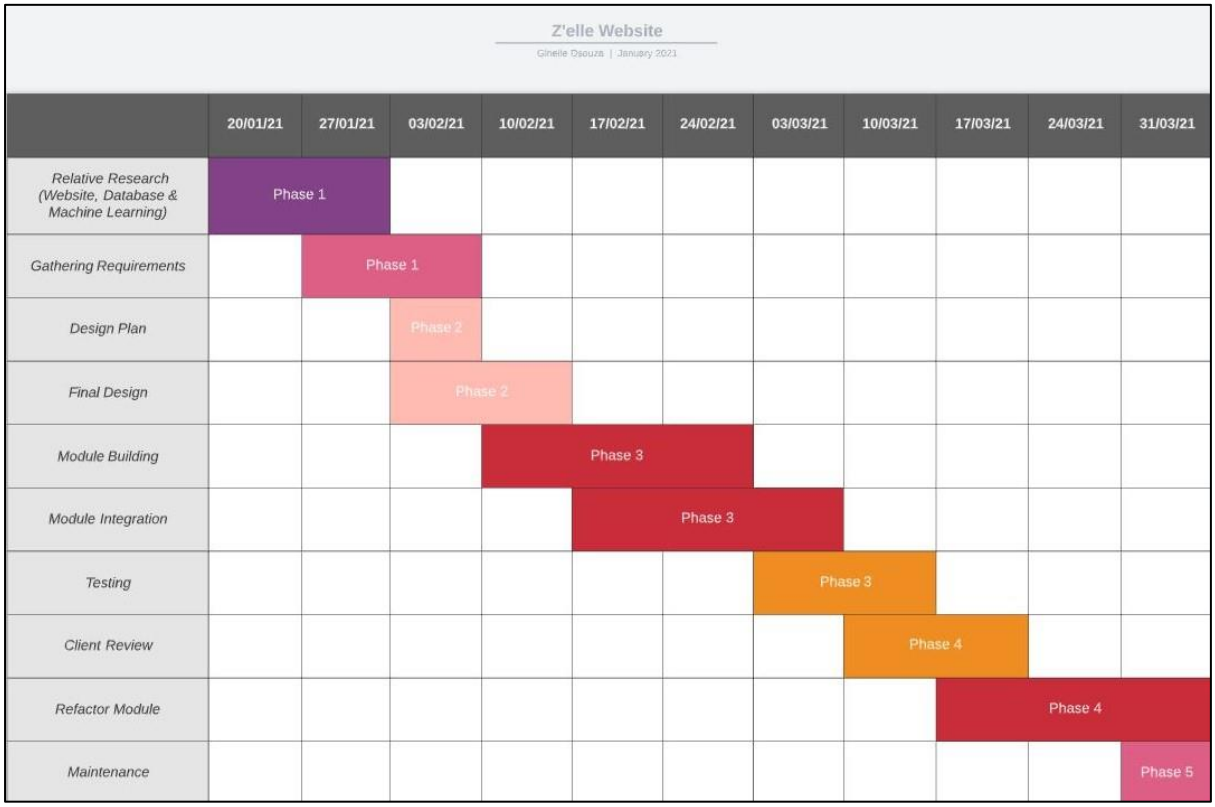
For calculating the Development costs, we need to follow the below mentioned steps:

1. Establish a framework to outline the parameters of the analysis
2. Identify costs and benefits so they can be categorized by type, and intent
3. Calculate costs and benefits across the assumed life of a project or initiative
4. Compare cost and benefits using aggregate information
5. Analyze results and make an informed, final recommendation

2.3 System planning and schedule

2.3.1 Gantt Chart

A Gantt chart is a project management tool assisting in the planning and scheduling of projects of all sizes, although they are particularly useful for simplifying complex projects. Project management timelines and tasks are converted into a horizontal bar chart, showing start and end dates, as well as dependencies, scheduling and deadlines, including how much of the task is completed per stage and who is the task owner.



CHAPTER 3

SYSTEM DESIGN

3.1 Software Requirement Specification

A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It also describes the functionality the product needs to fulfill all stakeholders (business, users) needs. SRS document is an agreement between the developer and the customer covering the functional and non-functional requirements of the software to be developed. SRS contains a contract between the customer and the developer. This SRS document is used for verifying whether all the functional and nonfunctional requirements specified in the SRS are implemented in the product. The complete description of the functions to be performed by the software specified in the SRS will assist the potential users to determine if the software specified meets their needs or how the software must be modified to meet their needs.

3.1.1. Introduction

- Processor: Intel core i5 or above
- Hard Disk: 10 GB or above.
- RAM: 4 GB or above.

3.1.2. Selection of Technology/Specific Requirement

Software

- Operating System: Windows
- Deployment server: Flask
- Front End: HTML, CSS, JavaScript
- Back End: PostgreSQL

Tools

- Database Server: Amazon Relational Database (Amazon RDS)
- Web Crawler: Beautiful Soup
- Data Visualization: Google Data Studio
- Data Prediction: Scikit-Learn - Random Forest (Machine Learning)

3.2 Methodologies Adapted

The methodology adapted for software development is **iterative software development** method. The Iterative model does not need a full list of requirements before beginning the project. The development process starts with the requirements of the functional part, which can be expanded later. The process is repetitive and allows new versions of the product for every cycle. Every iteration includes the development of a separate component of the system which is added to the functional developed earlier.

Advantages

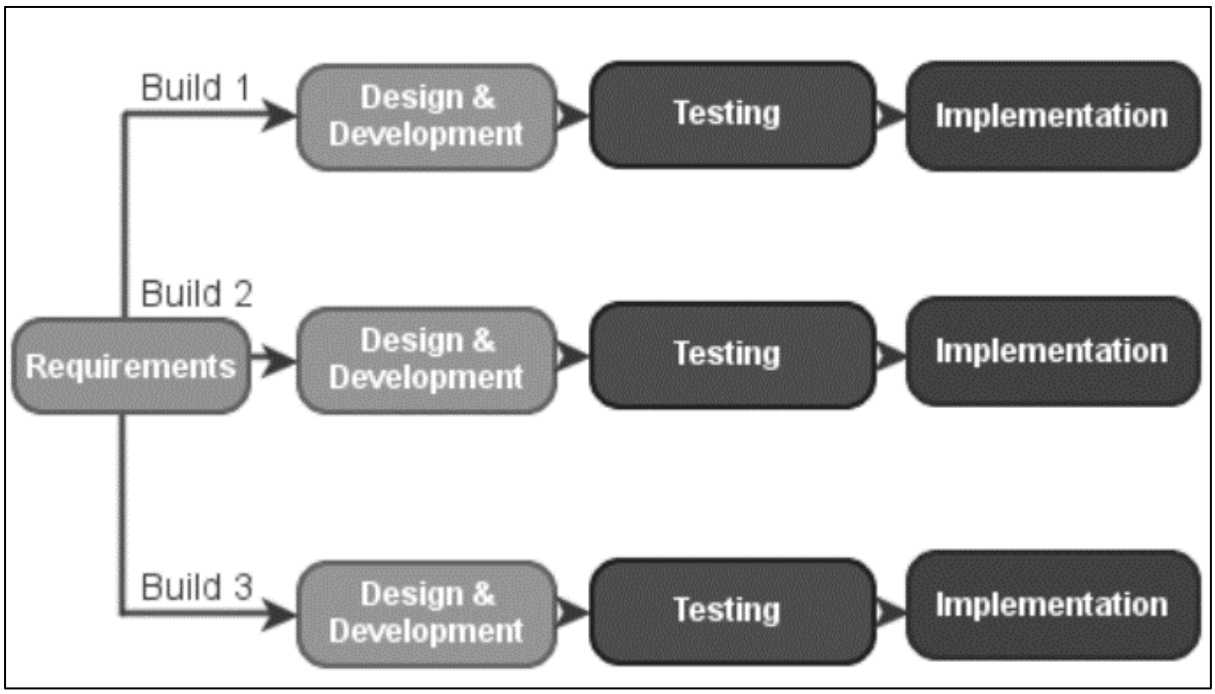
- It is easier to control the risks as high-risk tasks are completed first.
- The progress is easily measurable.
- Problems and risks defined within one iteration can be prevented in the next sprints.

Disadvantages

- Iterative model requires more resources than the waterfall model.
- The process is difficult to manage.
- The risks may not be completely determined even at the final stage of the project.

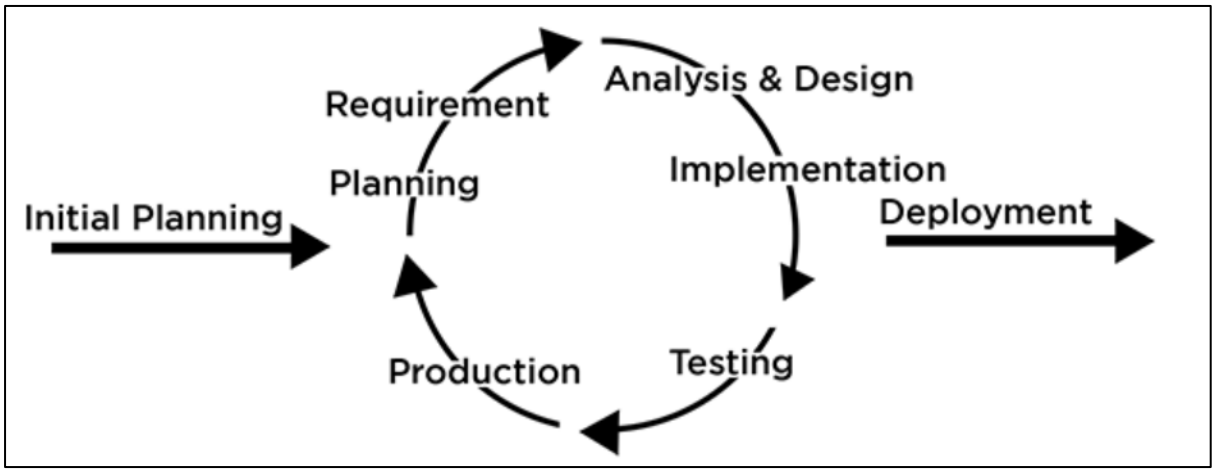
3.3 Detailed life Cycle of the Project

Iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental). Iterative and Incremental development is a combination of both iterative design or iterative method and incremental build model for development. "During software development, more than one iteration of the software development cycle may be in progress at the same time." This process may be described as an "**evolutionary acquisition**" or "**incremental build**" approach.



Every iteration involves several processes that is taken up across teams these process/workflows are as follows:

- Planning
- Requirement
- Analysis and Design
- Implementation
- Testing
- Production
- Deployment



Planning

Planning is used to map out the particular requirements given by the client. These requirements stated are closely noted and analyzed by the company. They can either be hardware or software requirements.

Requirement

Requirement Analysis, also known as Requirement Engineering, is the process of defining user expectations for a new software being built or modified.

Analysis and Design

When the planning stage is done, it is time to determine the business logic of the project. Business logic enables communication between the database and the end-users.

Implementation

This stage of the iterative design approach is clear: start the development process based on the set requirements and users' feedback. Before the beginning of each iteration, the development team negotiates the tasks' priority and details with the client. All the changes to the code are usually uploaded to the staging site first.

Testing

When the development team is done with coding, it is necessary to perform testing to find and fix all the bugs and errors. Testing the product is not enough, QA engineers should also pay close attention to user documentation and test it as well.

Production

Production is a process of combining various several modules in order to make something for the end user (output). It is the act of creating an output, a good or service which has value and contributes to the utility of individuals.

Deployment

Software deployment is all of the activities that make a software system available for use. The general deployment process consists of several interrelated activities with possible transitions between them. These activities can occur at the producer side or at the consumer side or both. Because every software system is unique, the precise processes or procedures within each activity can hardly be defined. Therefore, "deployment" should be interpreted as a general process that has to be customized according to specific requirements or characteristics

3.3.1 Modules

Analytic Module

Analytics is the systematic computational analysis of data or statistics. It is used for the discovery, interpretation, and communication of meaningful patterns in data. It also entails applying data patterns towards effective decision-making. It can be valuable in areas rich with recorded information; analytics relies on the simultaneous application of statistics, computer programming and operations research to quantify performance. Z'elle provides analysis to cater to our clients as well as our end users.

Clients:

Z'elle records the action and desires of the users. This is done by the forms filled by the users within the website. Data extracted from these forms are visualized to show our customers, where the interest of the vast majority of the users lie. A detailed report can also be provided to our clients with in-depth analysis.

Users:

Several users are unaware of what may be the best technology for them. Through the analytics provided by Z'elle it is possible for users to view the choice among several users and probably take a call about the technology they were looking for through Z'elle.

Predictive Module

Predictive modeling, also called predictive analytics, is a mathematical process that seeks to predict future events or outcomes by analyzing patterns that are likely to forecast future results. The goal of predictive modeling is to answer this question: "Based on known past behavior, what is most likely to happen in the future? Two of the most widely used predictive modeling techniques are regression and neural networks. Companies can use predictive modeling to forecast events, customer behavior, as well as financial, economic, and market risks.

Current Trends (Technical) Module

Latest trends and the most happening technical news are generated by this website by the help of automated bots. A bot is a program designed to automate tasks. Typically, these tasks are simple, repetitive, and routine. So, a software bot can perform them quicker and more efficiently than a human could. Z'elle bots are designed to scarp the news from NDTV Gadgets and present our users with the best and most valuable information within current technological frontier. For this purpose, we have used Beautiful Soup. It is a Python library for getting data out of HTML, XML, and other markup languages. Beautiful Soup helps you pull particular content from a webpage, remove the HTML markup, and save the information. It is a tool for web scraping that helps you clean up and parse the documents you have pulled down from the web.

Content Based Recommendation Module

Recommender systems are active information filtering systems which personalize the information coming to a user based on his interests, relevance of the information etc. Recommender systems are used widely for recommending movies, articles, restaurants, places to visit, items to buy etc. A content-based recommender works with data that the user provides, either explicitly or implicitly. Based on that data, a user requirement is generated, which is then used to make suggestions to the user. As the user provides more inputs or takes actions on the recommendations, the engine becomes more and more accurate.

3.3.2 Object Oriented Analysis & Design Diagrams

Object-oriented techniques are thought to work well in situations in which complicated information systems are undergoing continuous maintenance, adaptation, and redesign. Object-oriented approaches use the industry standard for modeling object-oriented systems, called the unified modeling language (UML), to break down a system into a use case model.

Object-oriented programming differs from traditional procedural programming by examining objects that are part of a system. Each object is a computer representation of some actual thing or event. Objects may be customers, items, orders, and so on. Objects are represented by and grouped into classes that are optimal for reuse and maintainability. A class defines the set of shared attributes and behaviors found in each object in the class.

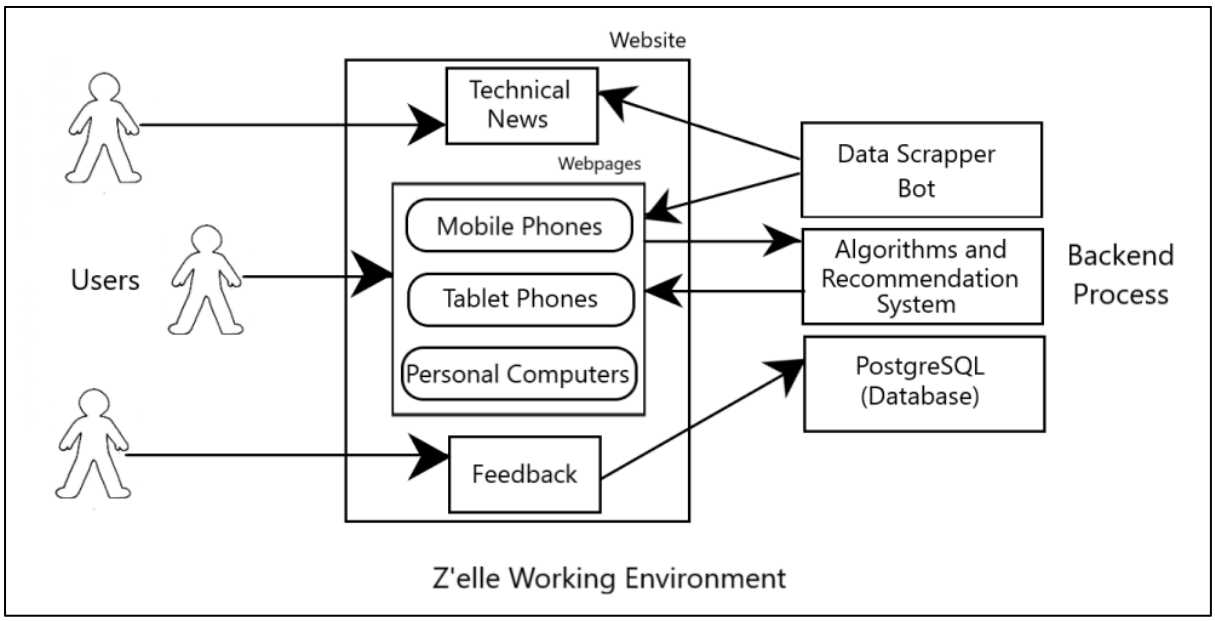
The phases in UML are similar to those in the SDLC. Since those two methods share rigid and exacting modeling, they happen in a slower, more deliberate pace than the phases of agile modeling. The analyst goes through problem and identification phases, an analysis phase, and a design phase

3.3.2.1 UML Diagram

1) Use Case Diagram

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. A use case diagram is usually simple. It does not show the detail of the use cases:

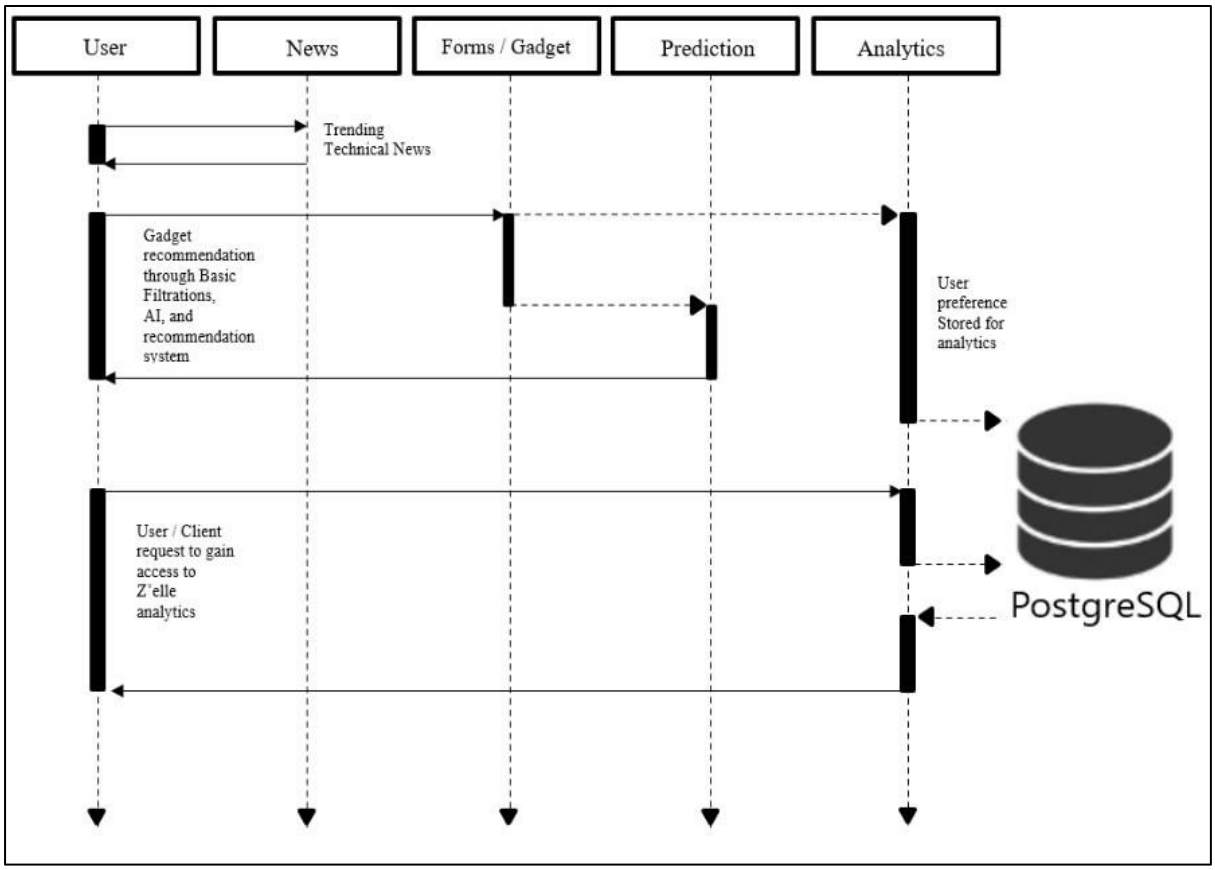
- It only summarizes some of the relationships between use cases, actors, and systems.
- It does not show the order in which steps are performed to achieve the goals of each use case.



2) Sequence Diagram

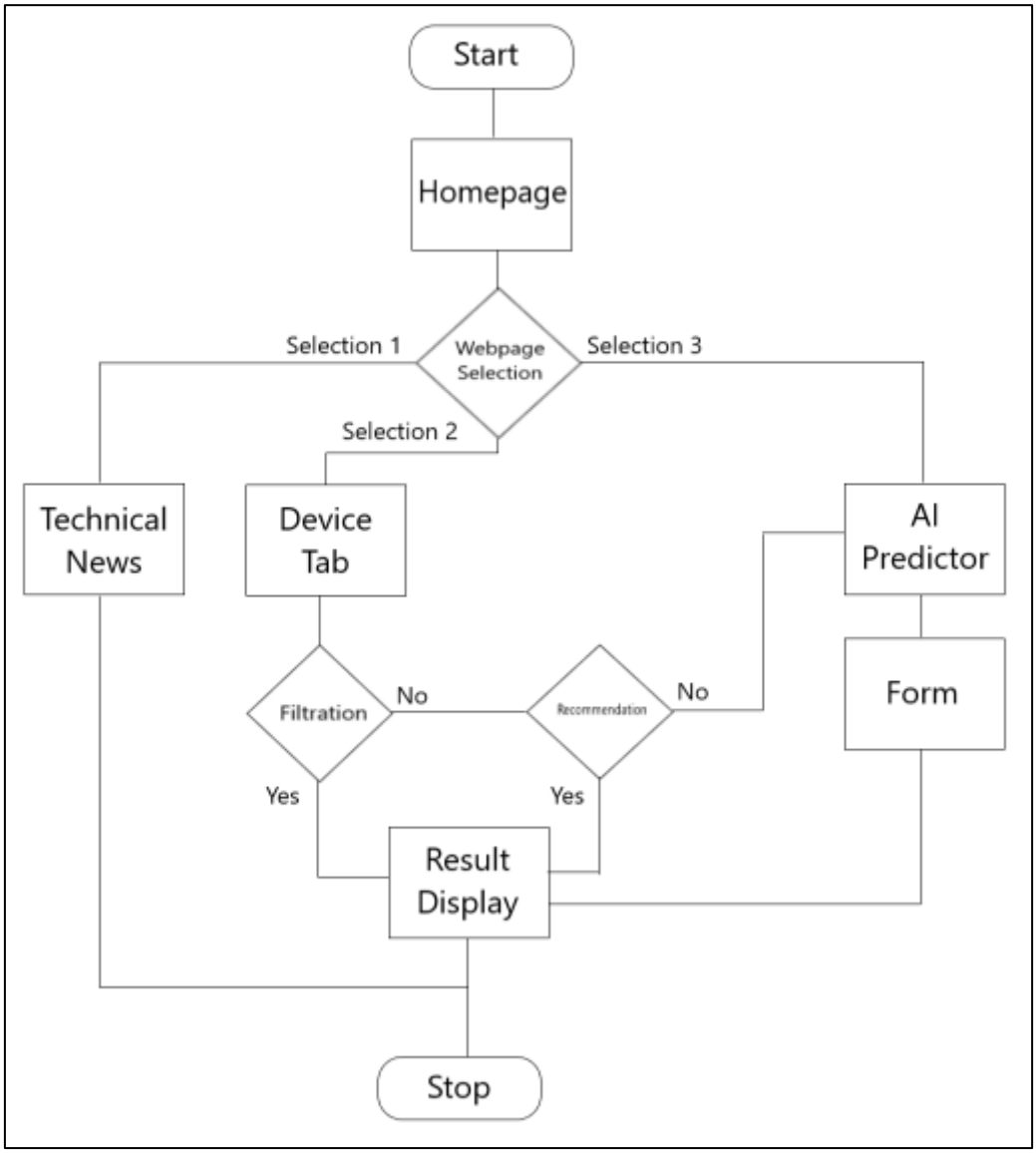
A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence Diagrams captures:

- The interaction that takes place in a collaboration that either realizes a use case or an operation (instance diagrams or generic diagrams)
- High-level interactions between user of the system and the system, between the system and other systems, or between subsystems (sometimes known as system sequence diagrams)



3) Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent.



3.3.3 Database Tables

- **Device Table**: The device database stores the information of all the scrapped gadget data done through web scrapping. This table will be used for analysis of gadgets, their price, review, and ratings over a period of time

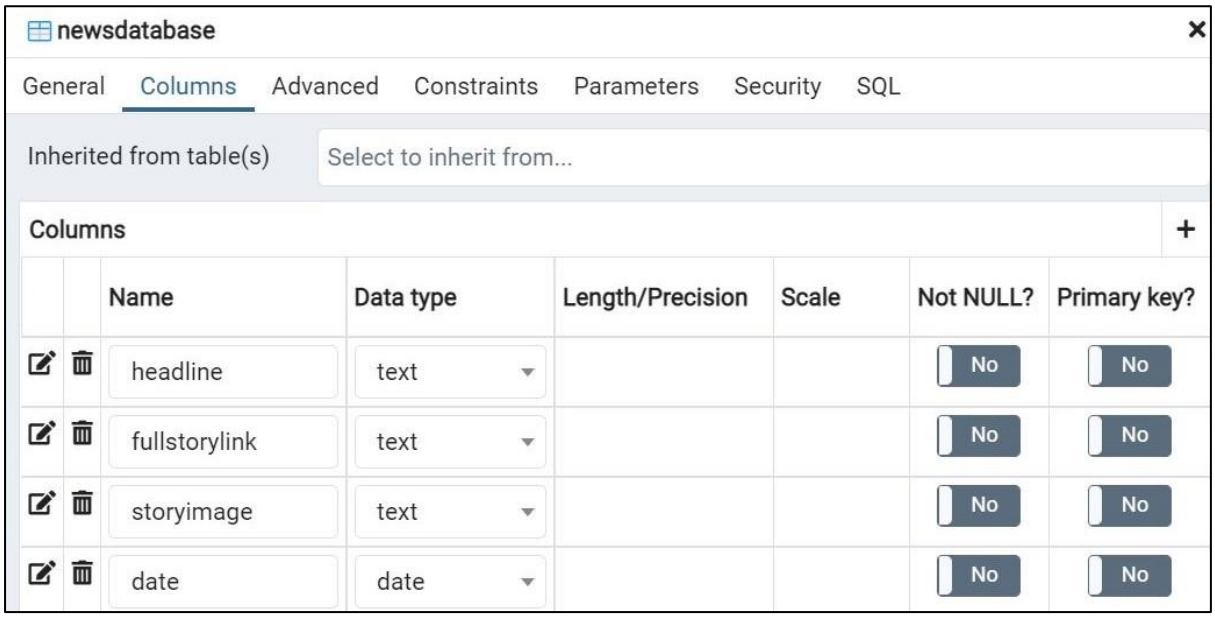
devicedatabase

General | Columns | Advanced | Constraints | Parameters | Security | SQL

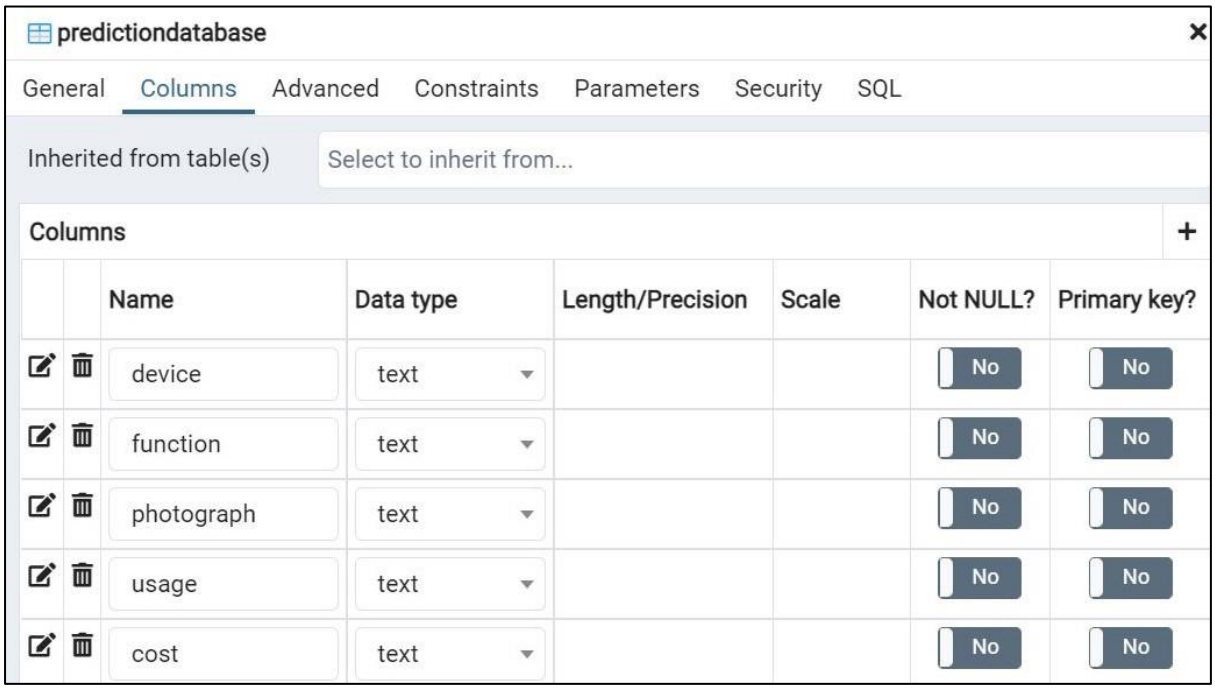
Columns +

		Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
✎	🗑	devicetype	text			No	No
✎	🗑	brand	text			No	No
✎	🗑	devicename	text			No	No
✎	🗑	deviceos	text			No	No
✎	🗑	deviceram	integer			No	No
✎	🗑	devicerom	integer			No	No
✎	🗑	devicebattery	integer			No	No
✎	🗑	deviceprice	integer			No	No
✎	🗑	rating	integer			No	No

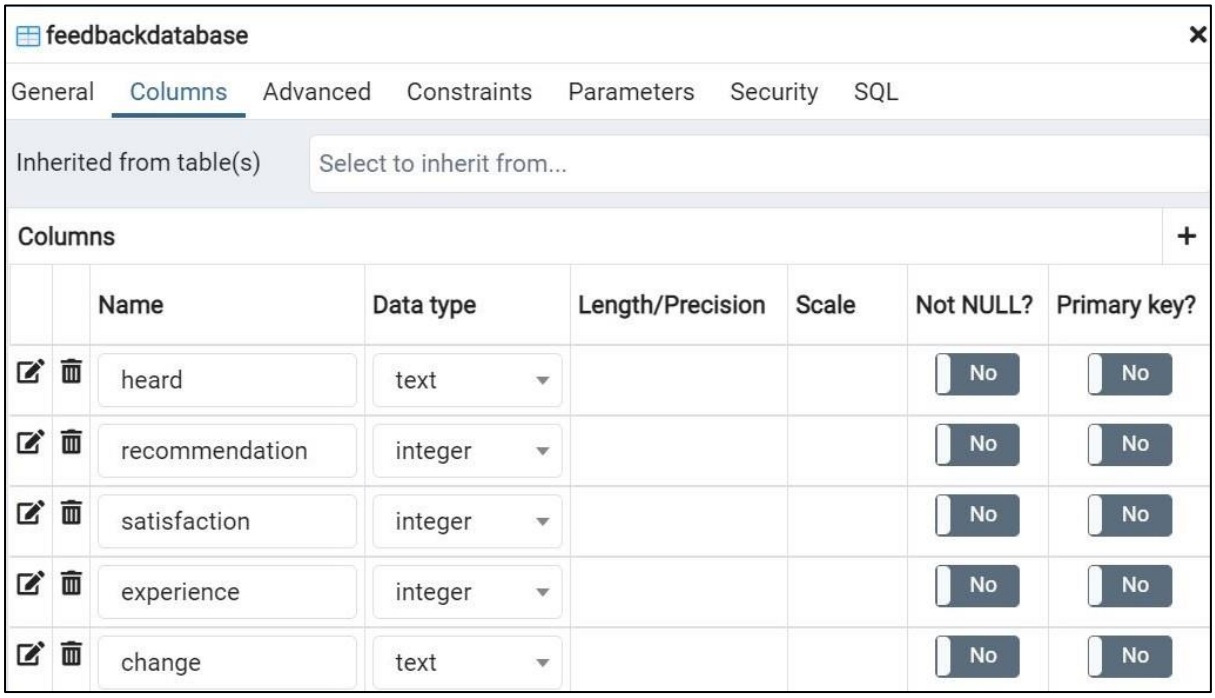
- **News Table**: The news database stores the information of all the scrapped trending news over time. This table will be used for analysis in the future.



- **Prediction Table**: The prediction database stores the information taken from the users while they fill out the AI prediction form. This data is stored such that Z'elle is able to provide a detailed analysis to their clients and users

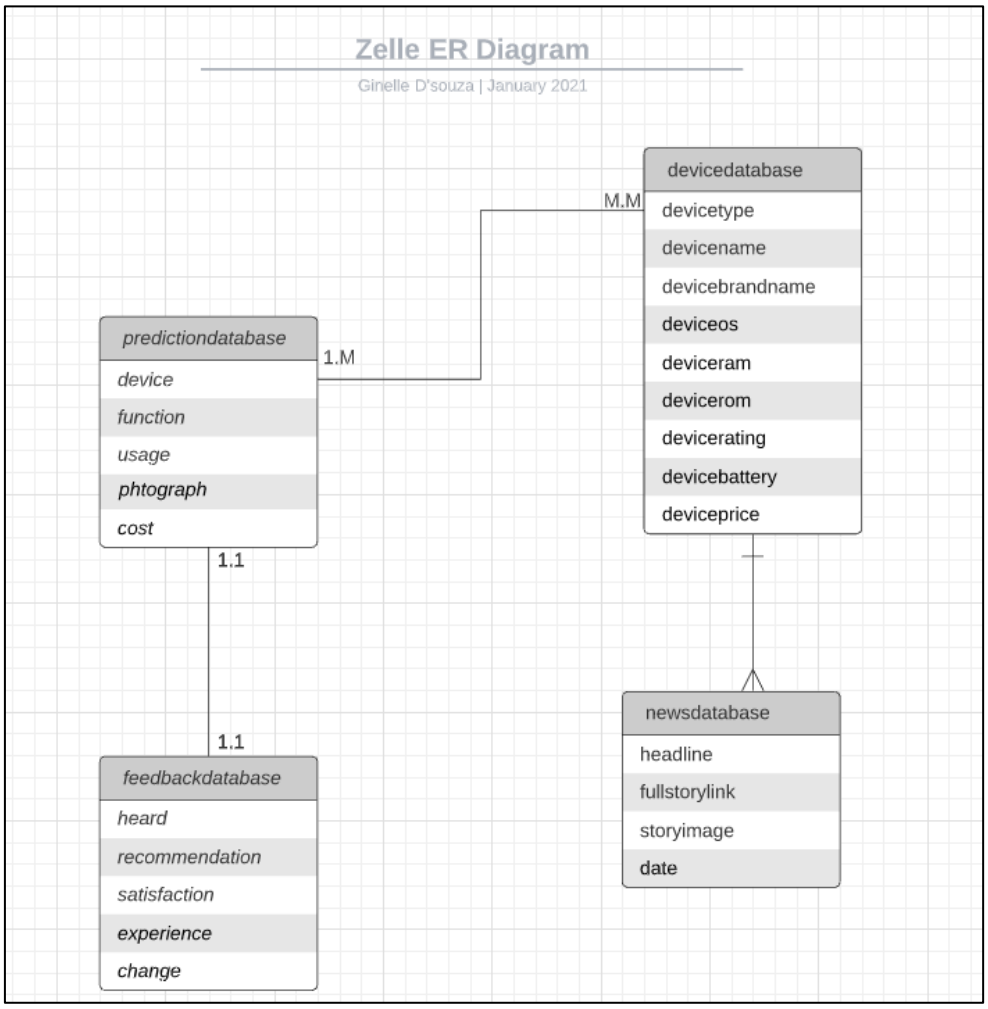


- **Feedback Table**: The feedback database stores the information taken from the users while they fill out the feedback form. This data is stored such that Z'elle is able to provide a detailed analysis to their clients and users



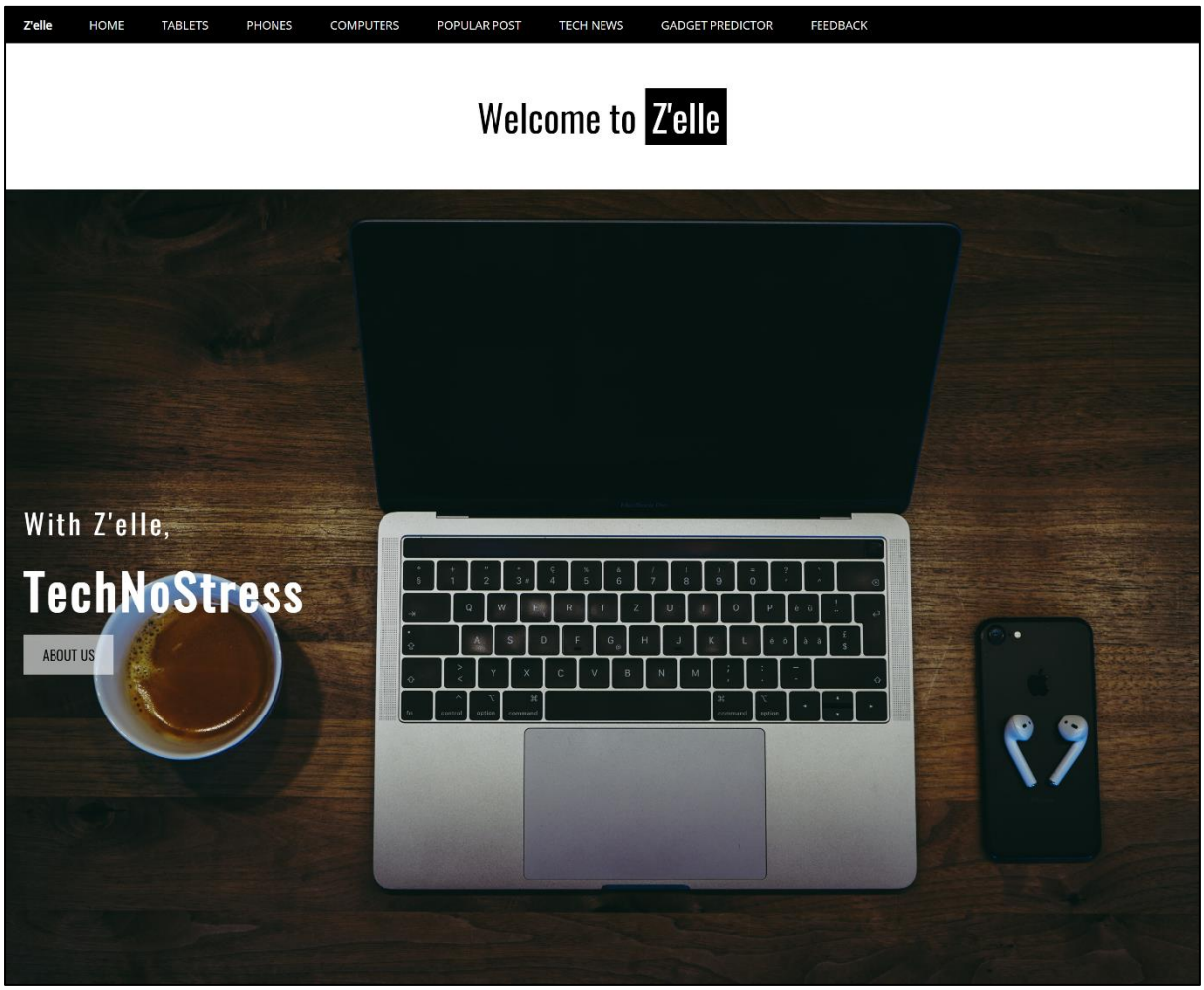
3.3.4 E R Diagram

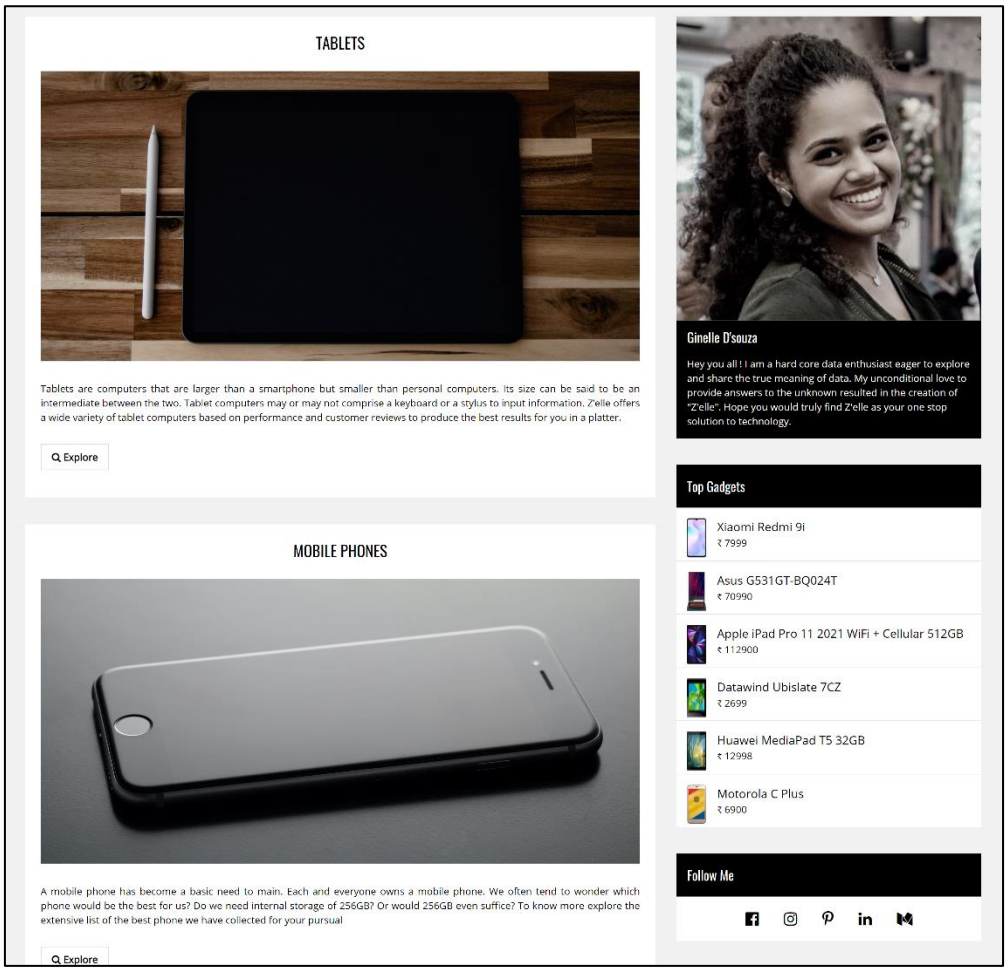
ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships. ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

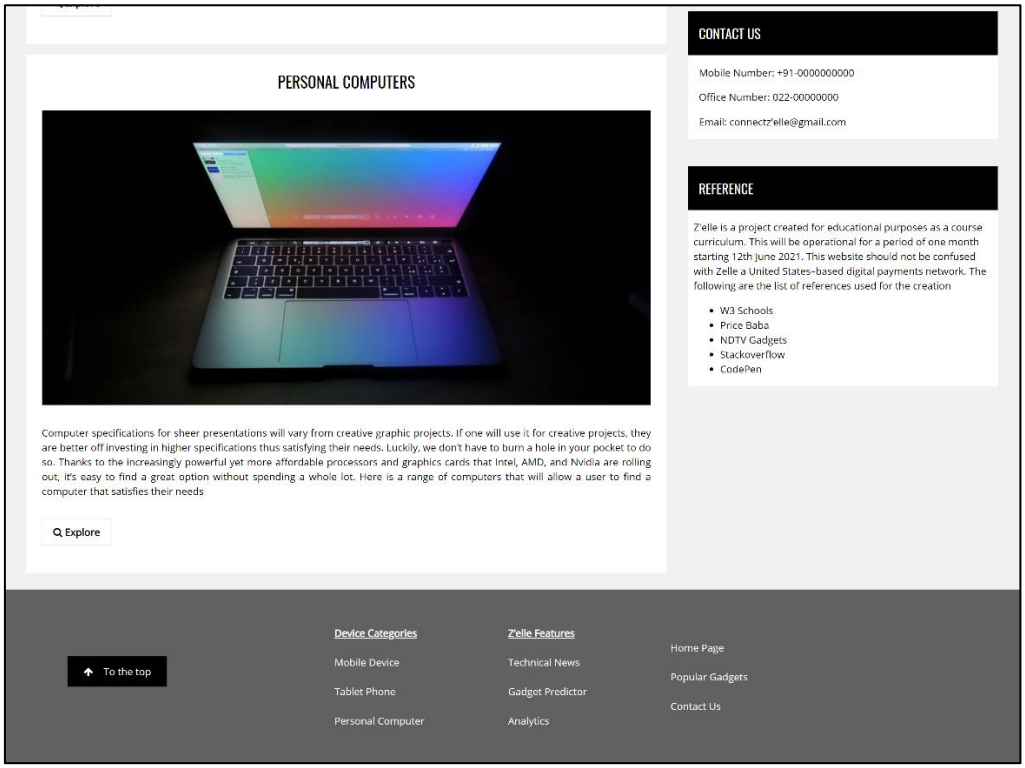


3.3.5 I/O Screen Layout

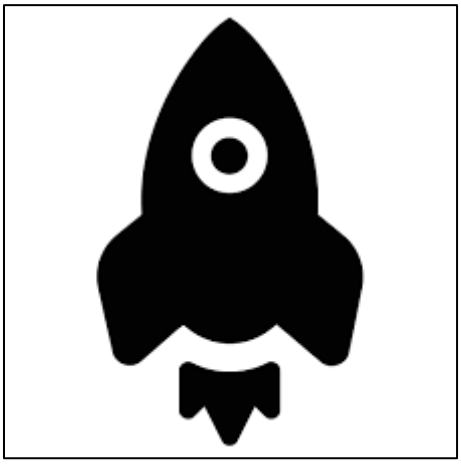
Homepage




TABLETS
Tablets are computers that are larger than a smartphone but smaller than personal computers. Its size can be said to be an intermediate between the two. Tablet computers may or may not comprise a keyboard or a stylus to input information. Z'elle offers a wide variety of tablet computers based on performance and customer reviews to produce the best results for you in a platter.
Explore
MOBILE PHONES
A mobile phone has become a basic need to main. Each and everyone owns a mobile phone. We often tend to wonder which phone would be the best for us? Do we need internal storage of 256GB? Or would 256GB even suffice? To know more explore the extensive list of the best phone we have collected for your pursual
Explore
Ginelle D'souza
Hey you all ! I am a hard core data enthusiast eager to explore and share the true meaning of data. My unconditional love to provide answers to the unknown resulted in the creation of "Z'elle". Hope you would truly find Z'elle as your one stop solution to technology.
Top Gadgets
Xiaomi Redmi 9i
₹ 7999
Asus G531GT-BQ024T
₹ 70990
Apple iPad Pro 11 2021 WiFi + Cellular 512GB
₹ 112900
Datawind Ubislate 7CZ
₹ 2699
Huawei MediaPad T5 32GB
₹ 12998
Motorola C Plus
₹ 6900
Follow Me


PERSONAL COMPUTERS
Computer specifications for sheer presentations will vary from creative graphic projects. If one will use it for creative projects, they are better off investing in higher specifications thus satisfying their needs. Luckily, we don't have to burn a hole in your pocket to do so. Thanks to the increasingly powerful yet more affordable processors and graphics cards that Intel, AMD, and Nvidia are rolling out, it's easy to find a great option without spending a whole lot. Here is a range of computers that will allow a user to find a computer that satisfies their needs
Explore
CONTACT US
Mobile Number: +91-0000000000
Office Number: 022-00000000
Email: connectz'elle@gmail.com
REFERENCE
Z'elle is a project created for educational purposes as a course curriculum. This will be operational for a period of one month starting 12th June 2021. This website should not be confused with Zelle a United States–based digital payments network. The following are the list of references used for the creation
W3 Schools
Price Baba
NDTV Gadgets
Stackoverflow
CodePen
To the top
Device Categories
Mobile Device
Tablet Phone
Personal Computer
Z'elle Features
Technical News
Gadget Predictor
Analytics
Home Page
Popular Gadgets
Contact Us

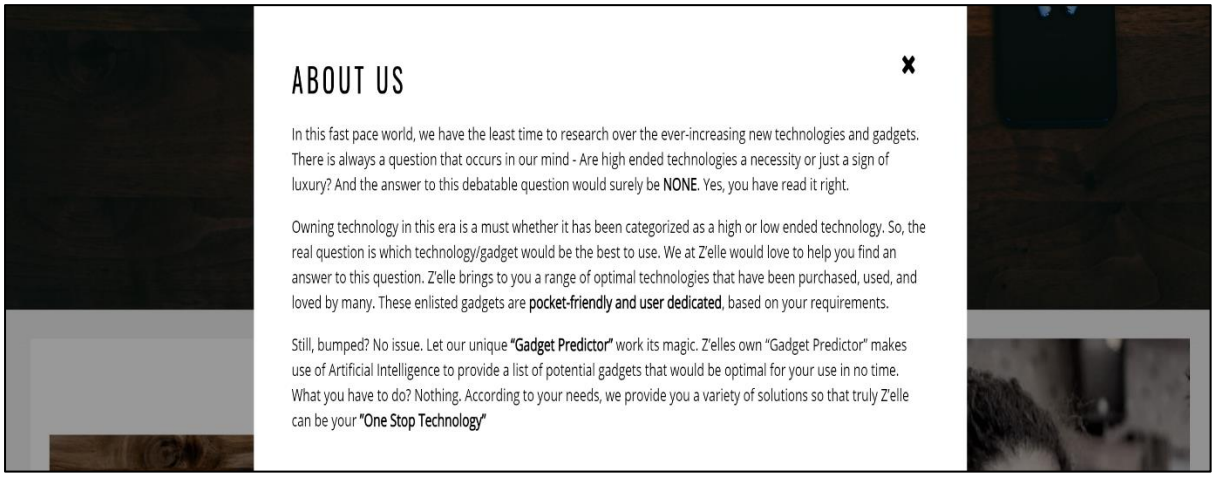
Z’elle Logo



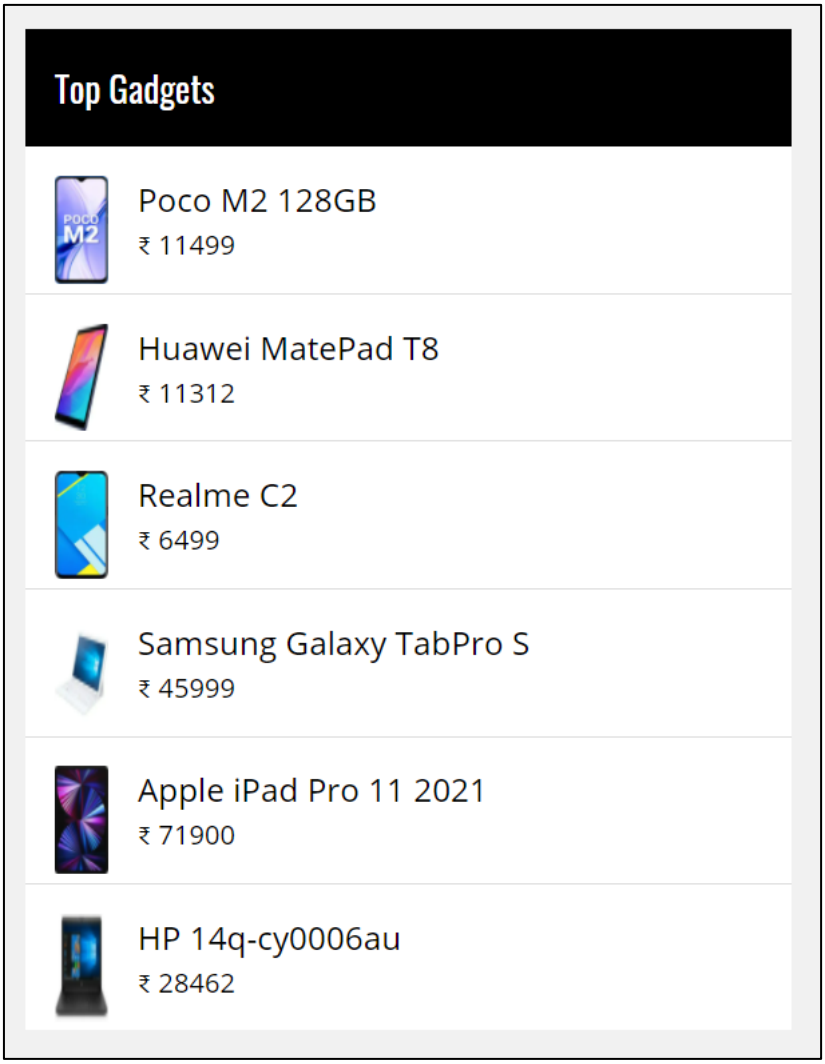
Header

Footer

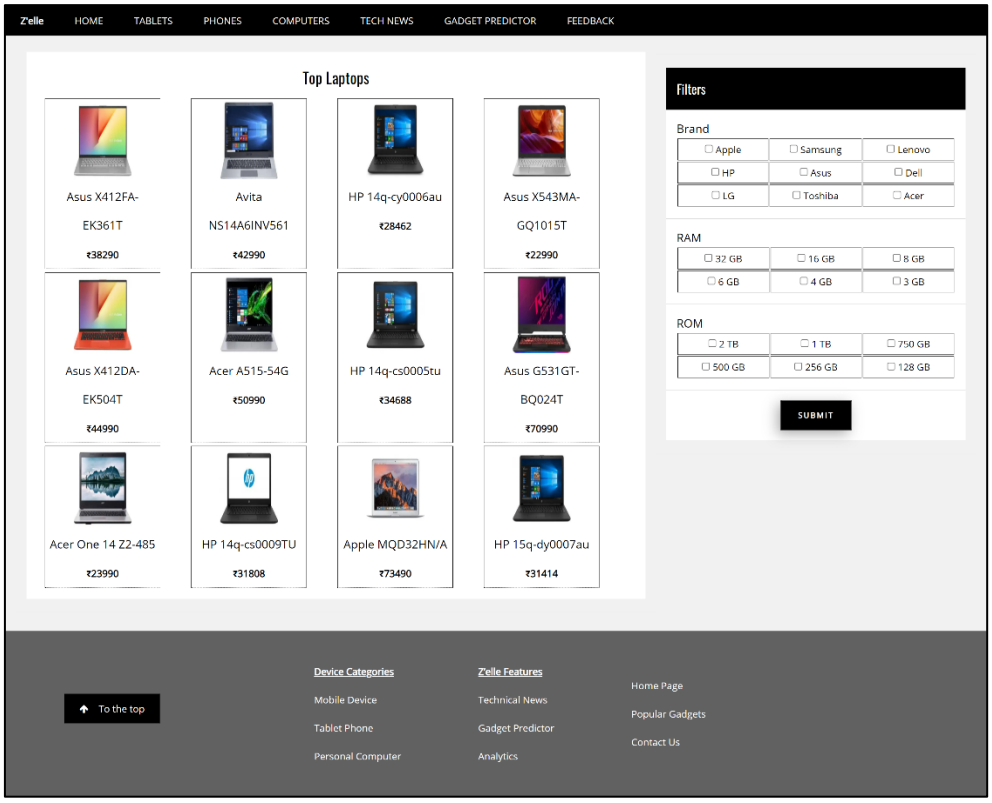
About Us



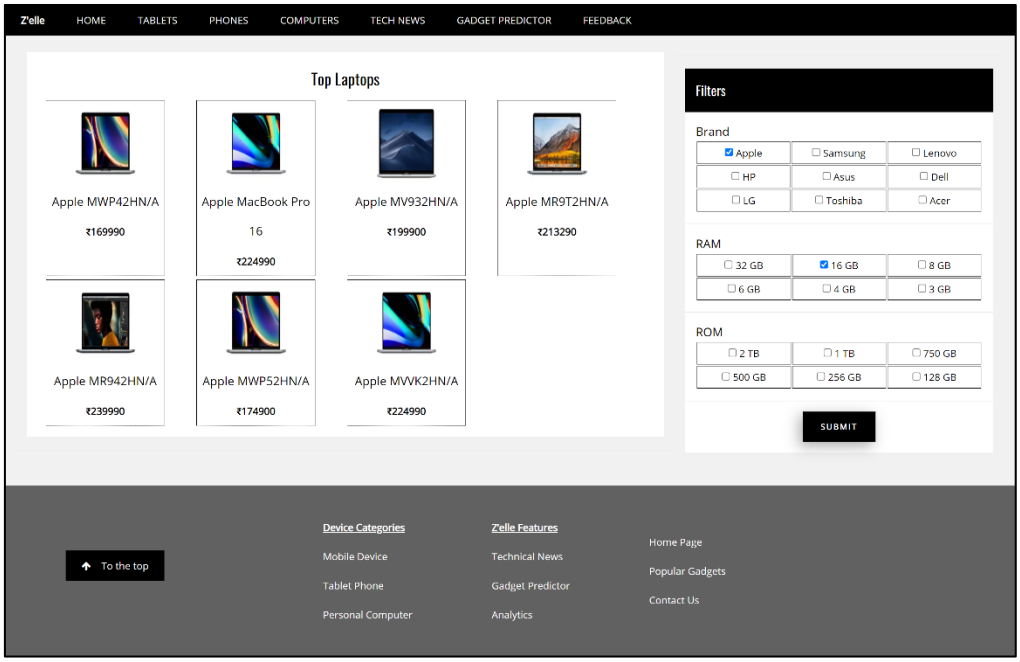
Trending Gadgets



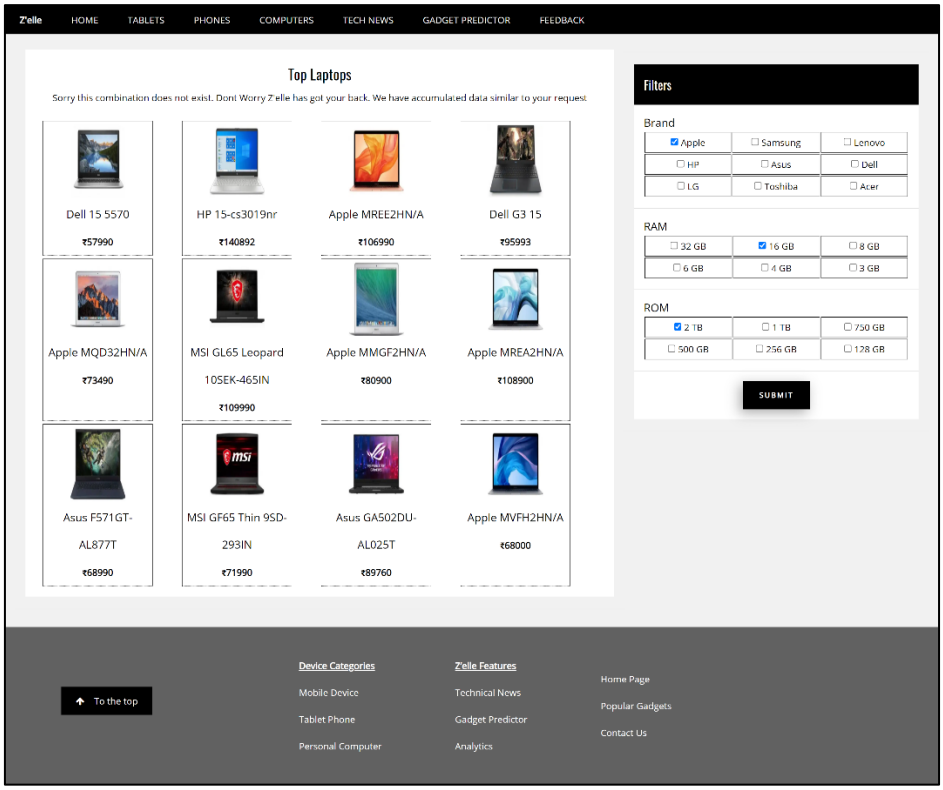
Personal Computers



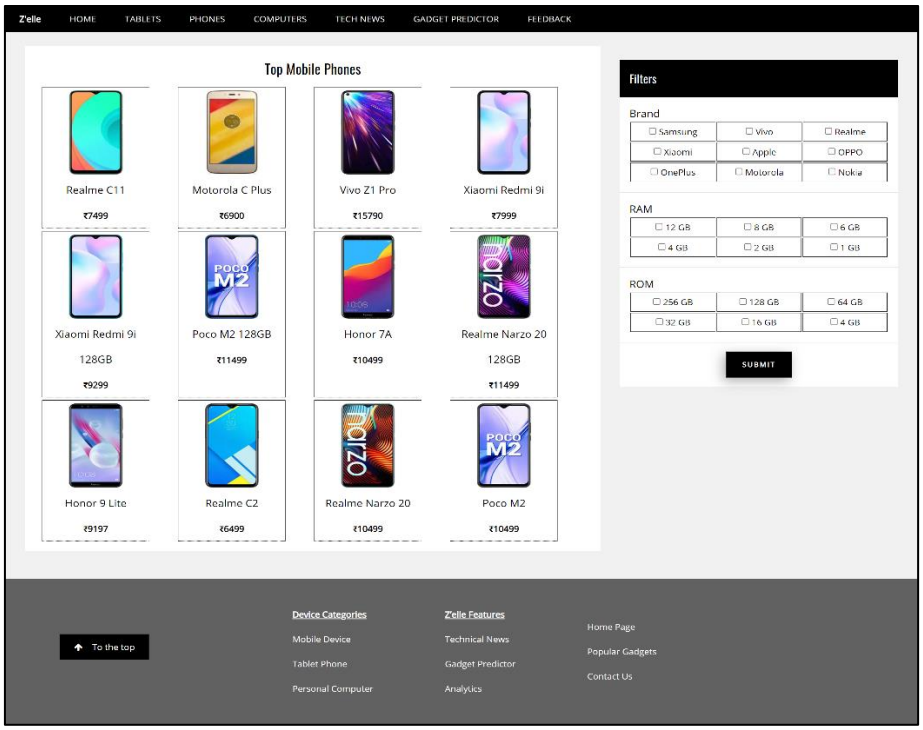
Basic Personal Computers Filtration



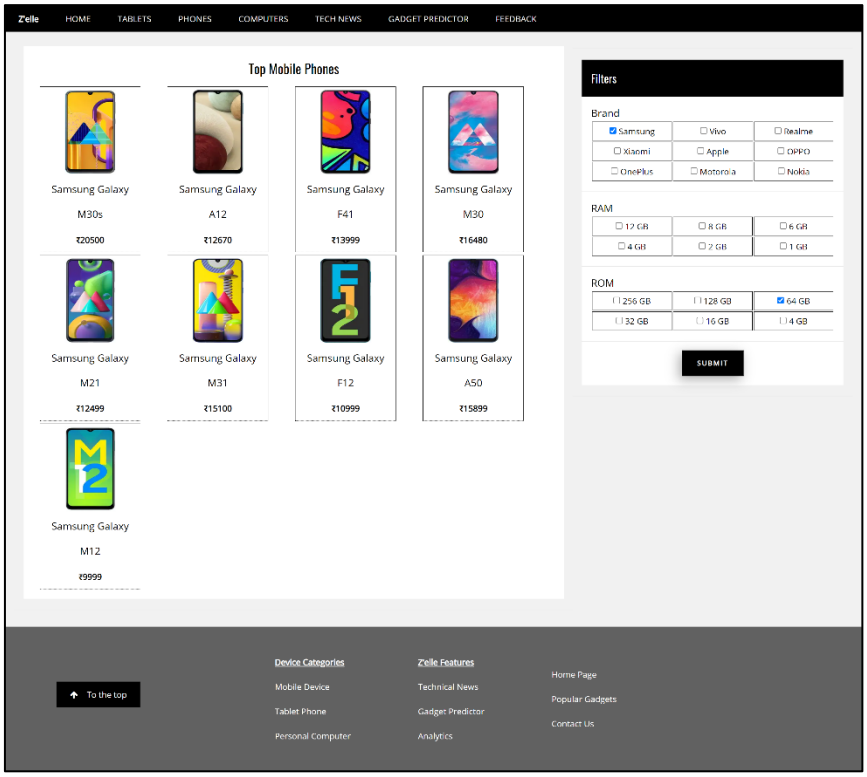
Personal Computers Recommendation



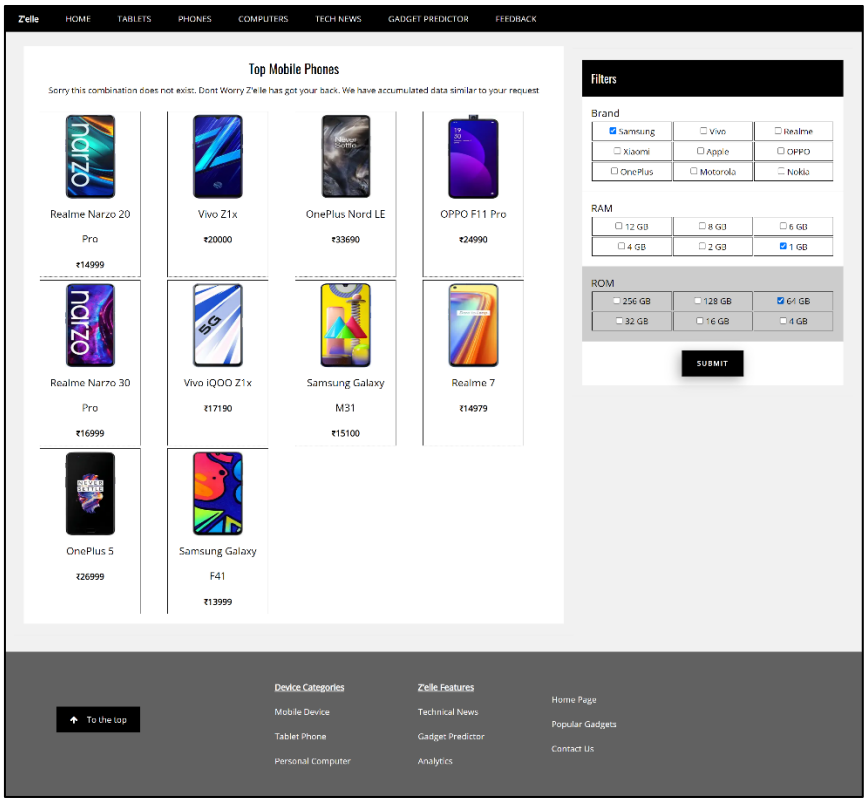
Mobile Phones



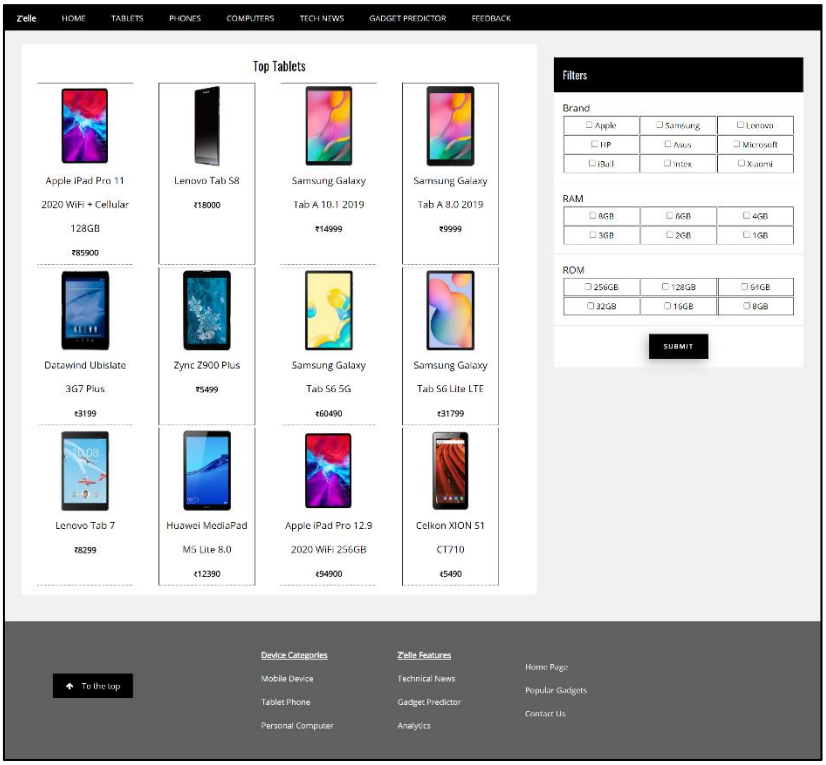
Basic Mobile Phones Filtration



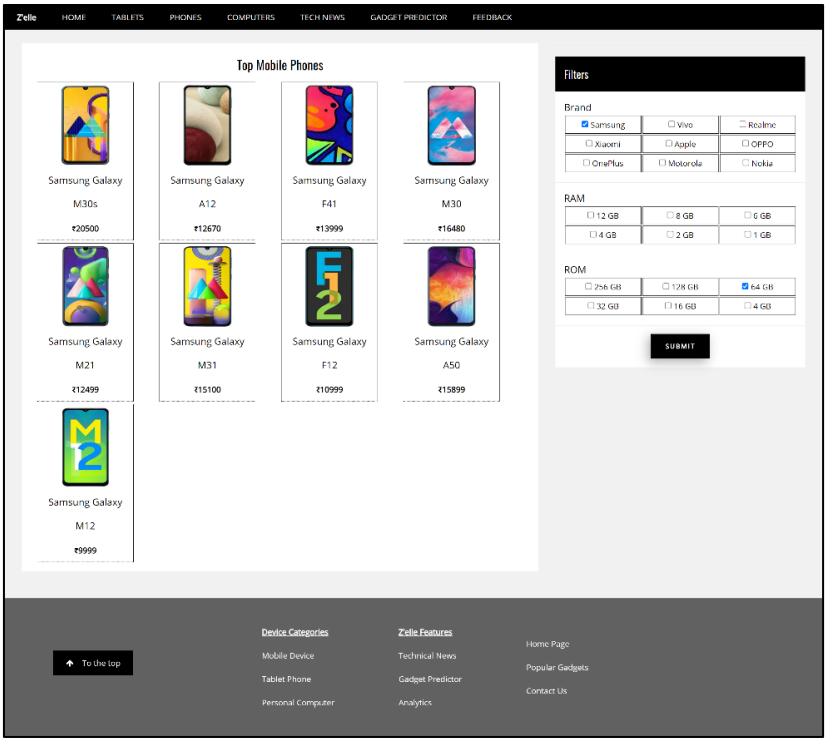
Mobile Phone Recommendation



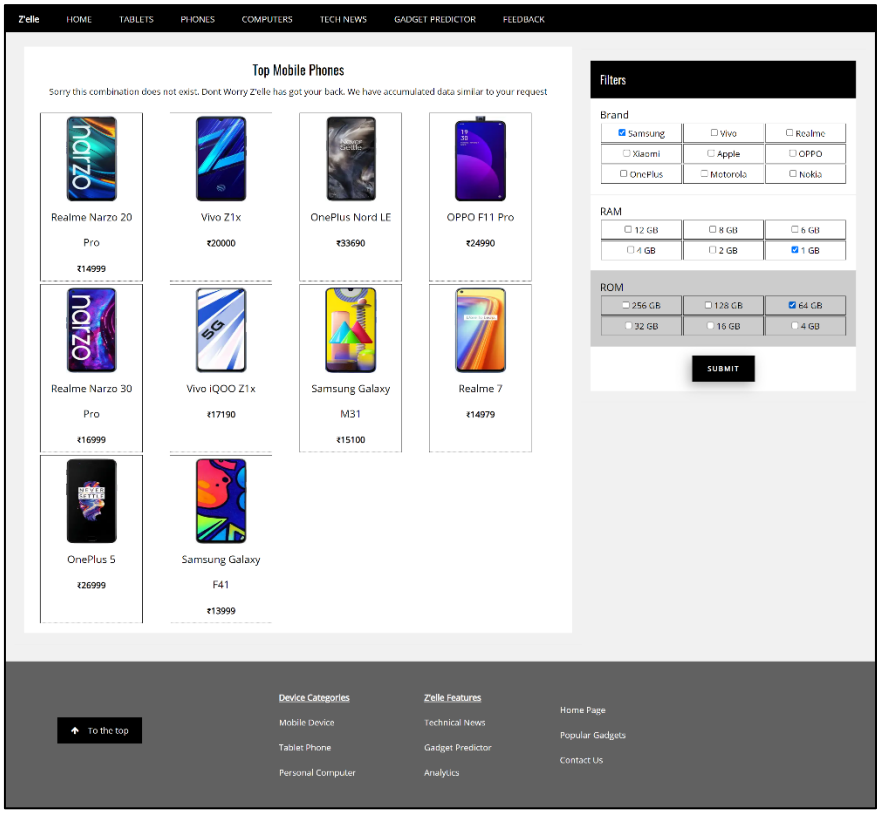
Tablet Phones



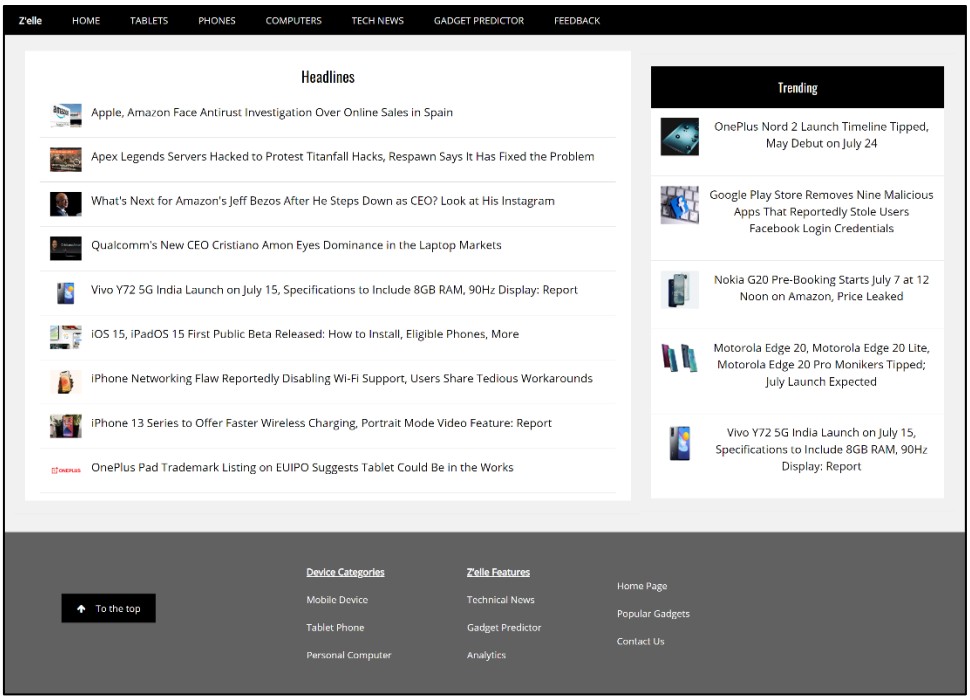
Basic Tablet Phones Filtration



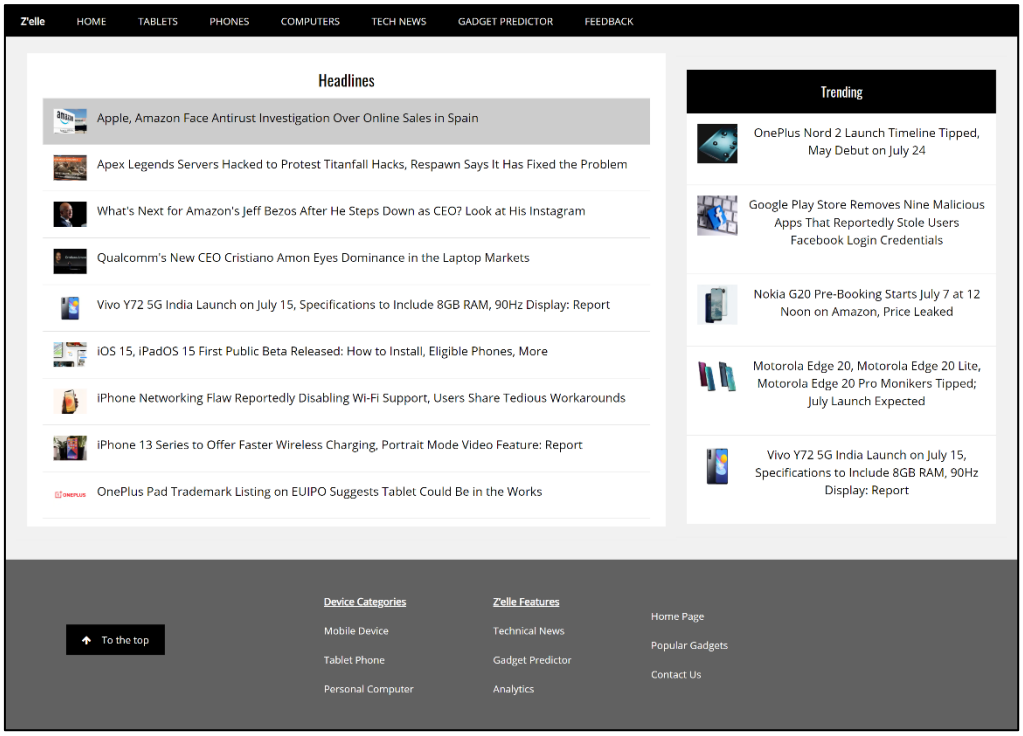
Tablet Phones Recommendation



Trending News

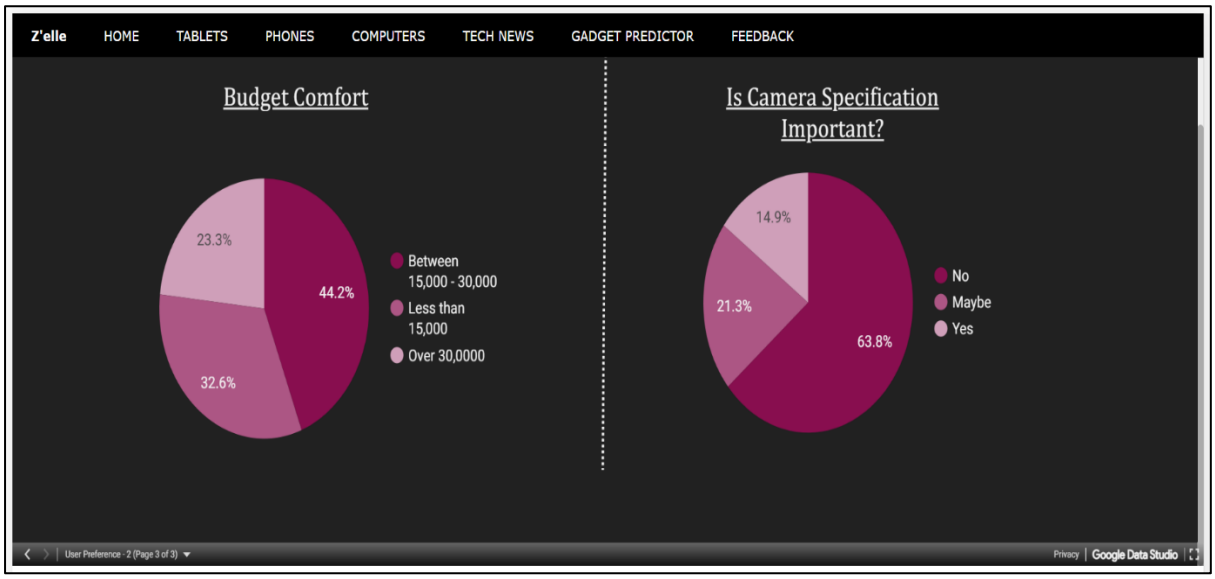


Detailed News Redirect



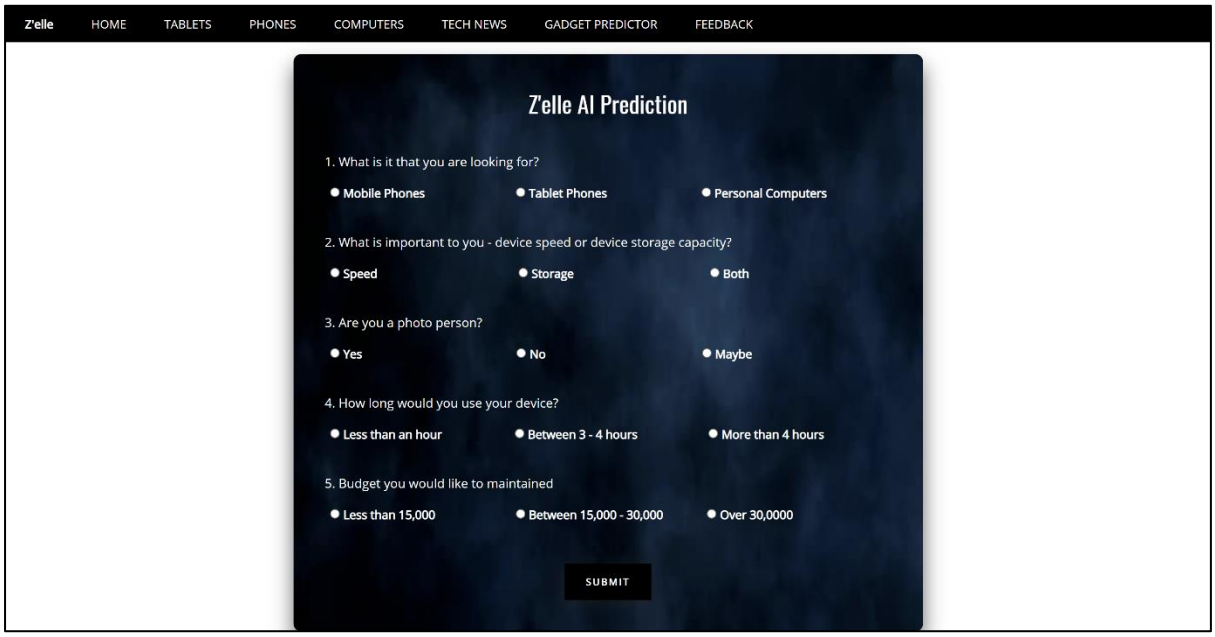
NDTV Gadget News

Zelle Analytics

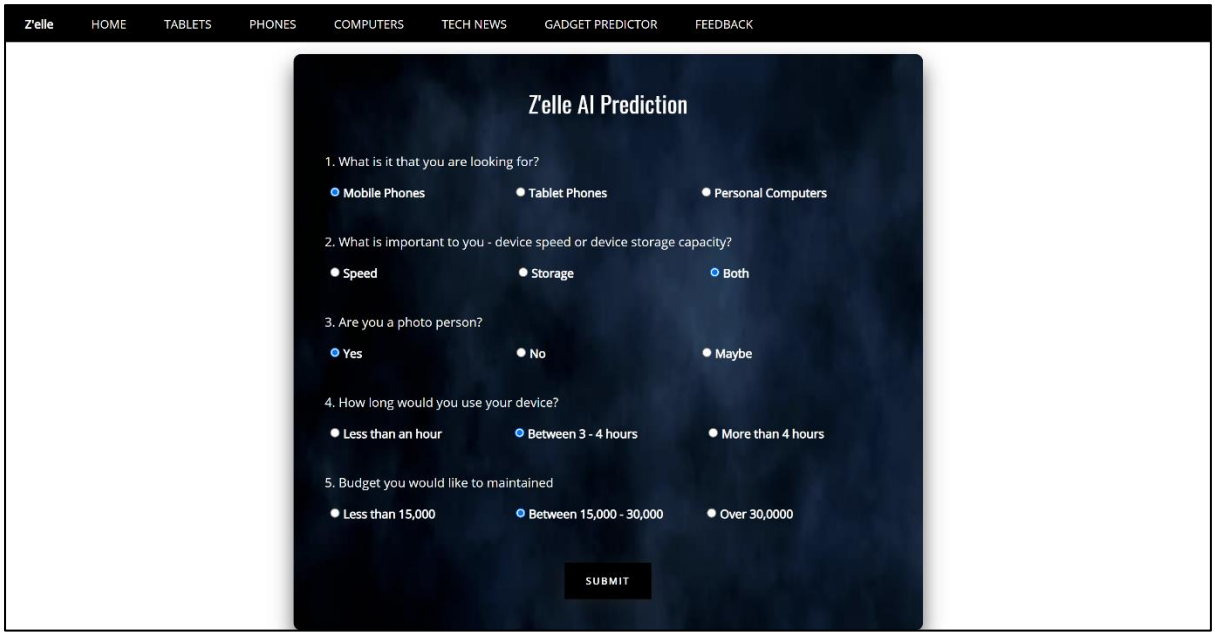


Gadget Prediction

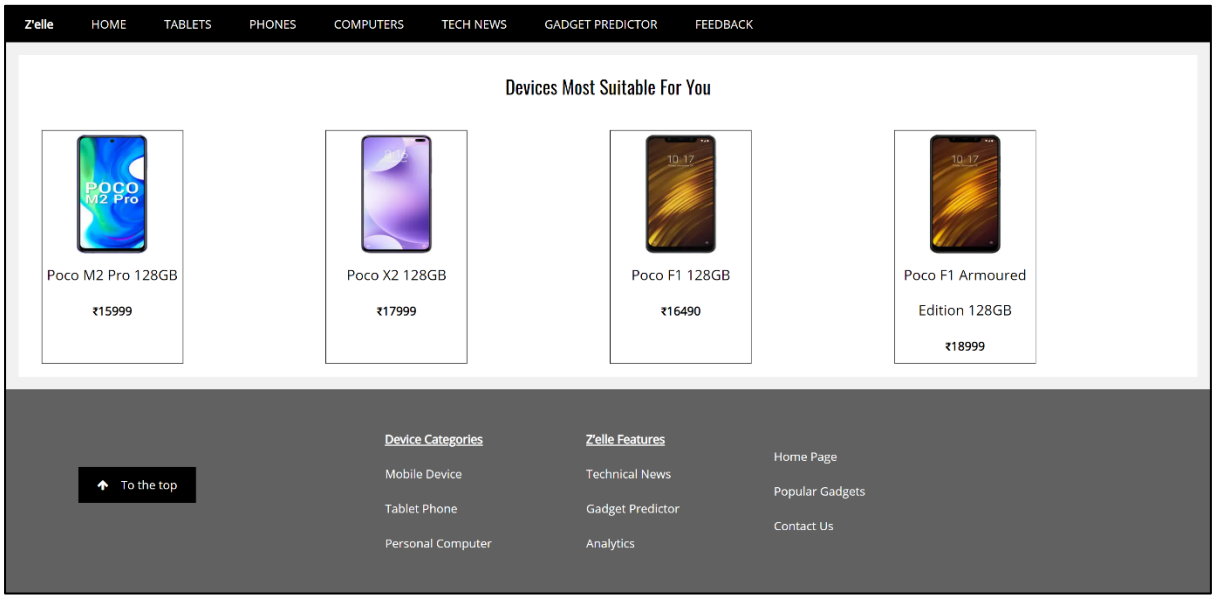
Predication Form



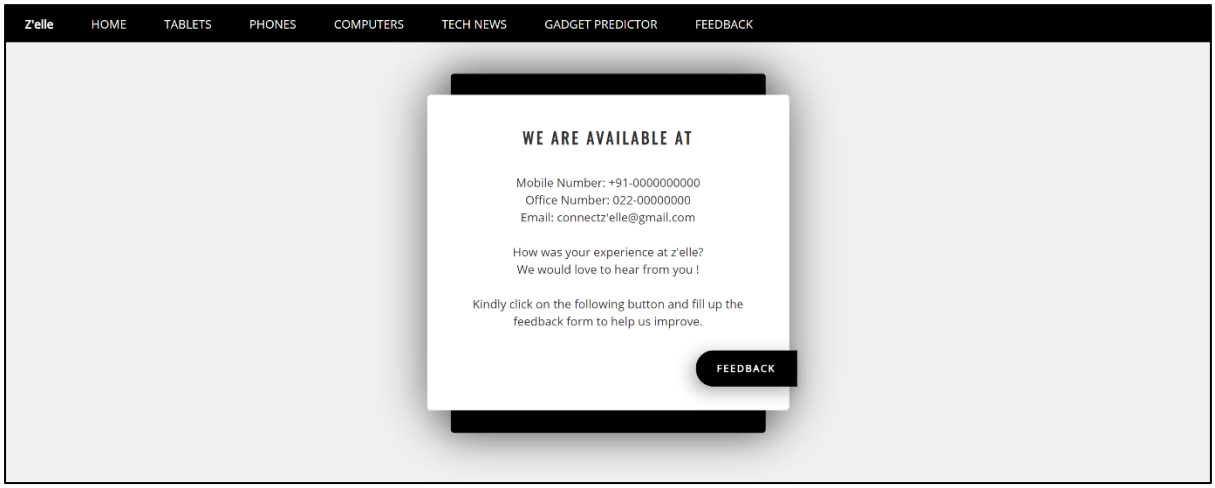
Form Selection



AI Recommendation

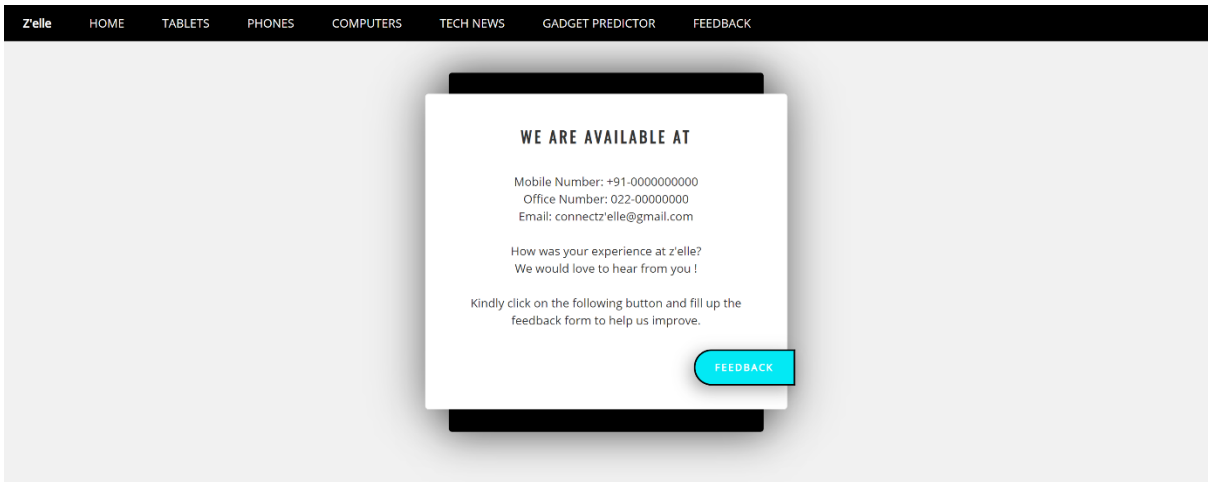


Contact Us

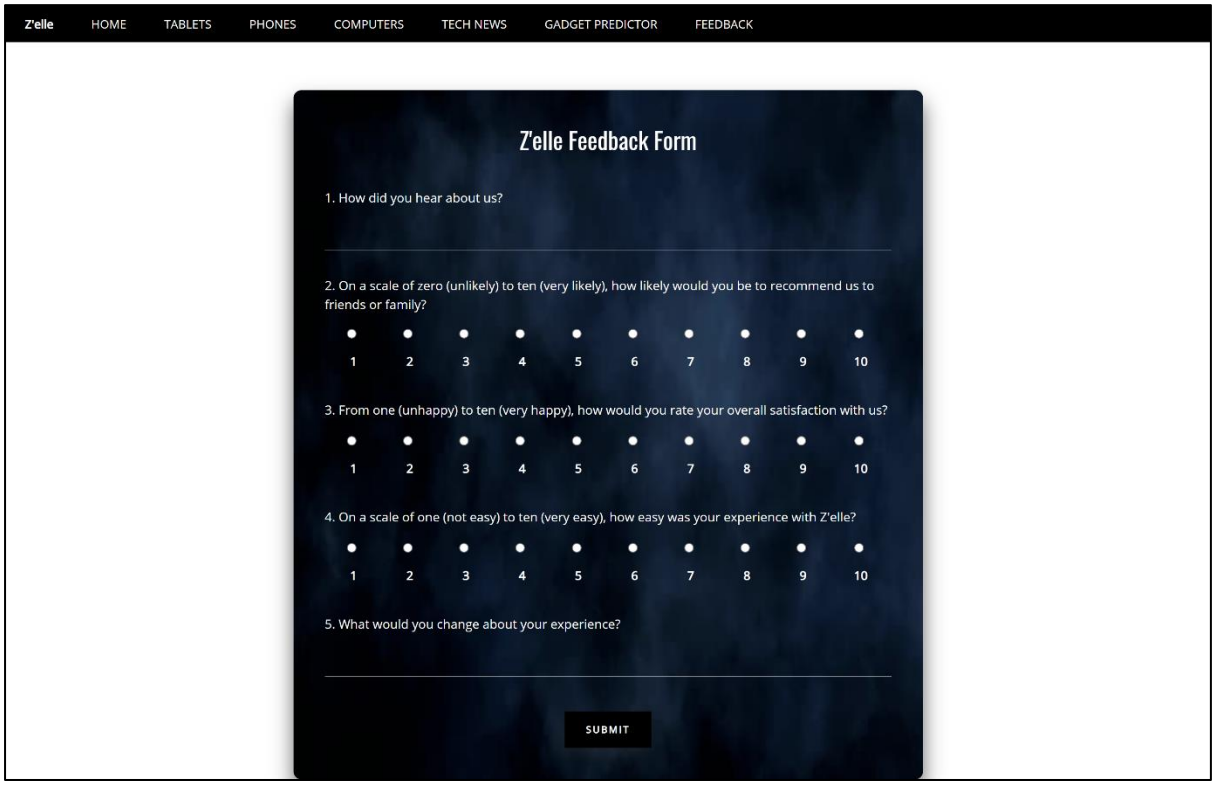


Feedback

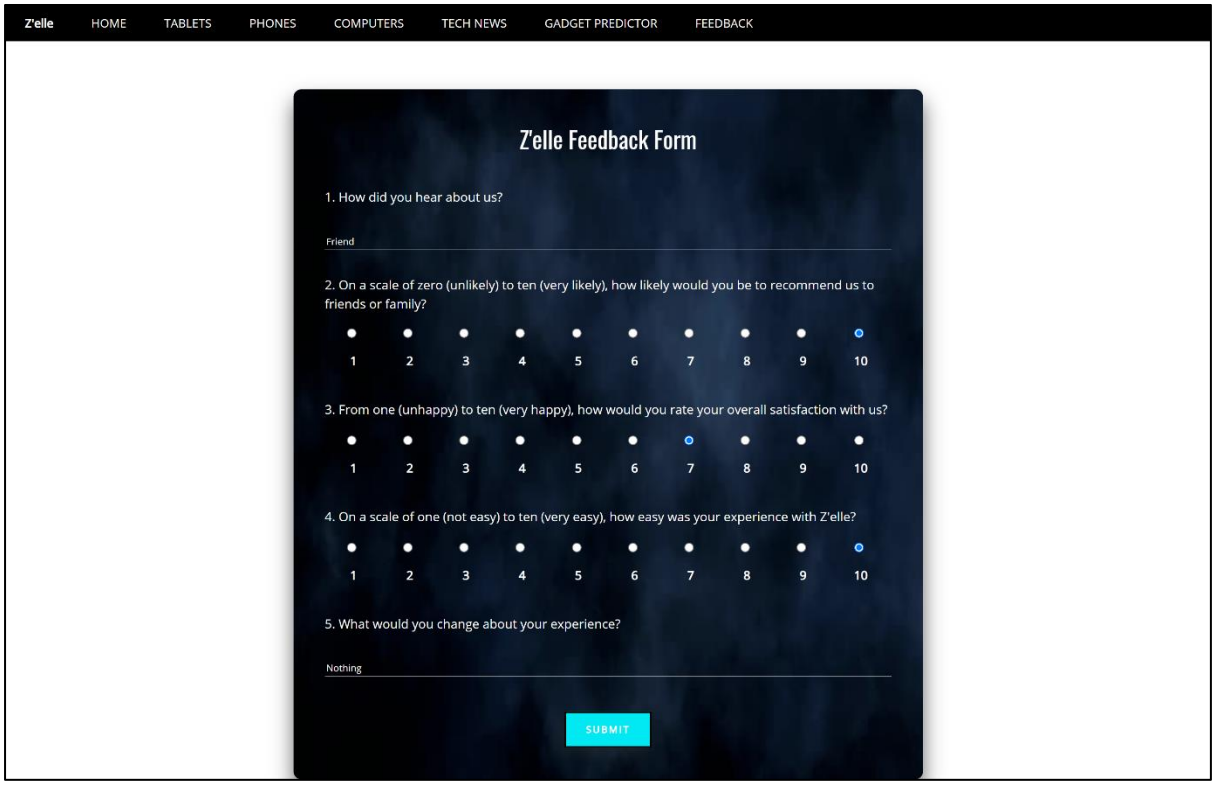
Form Redirect



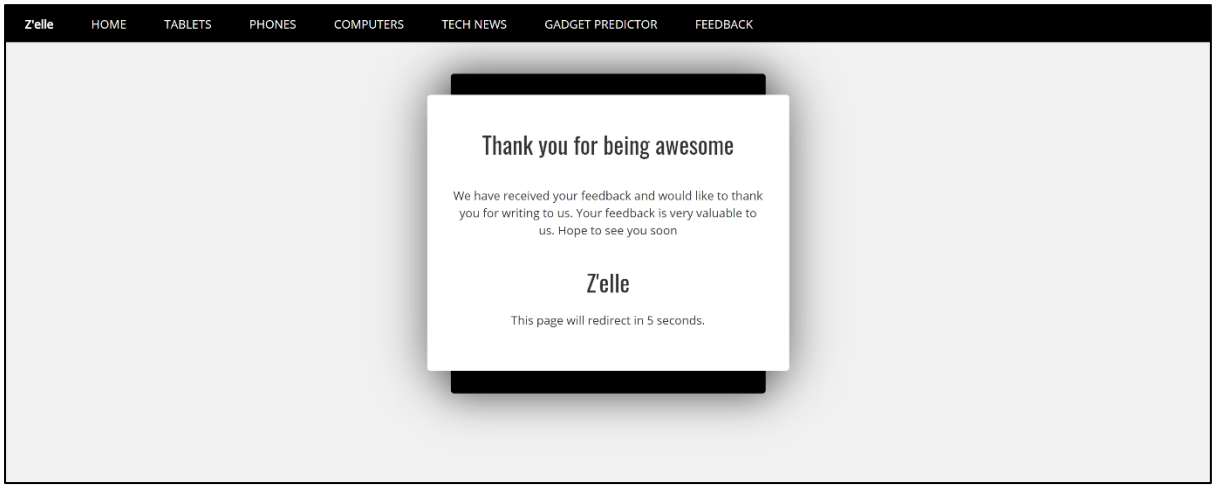
Feedback Form



Feedback Form Selection



Post Feedback Form



CHAPTER 4

TESTING

4.1 Methodologies Used For Testing

Software testing methodologies are the different approaches and ways of ensuring that a software application in particular is fully tested. Software testing methodologies encompass everything from unit testing individual modules, integration testing an entire system to specialized forms of testing such as security and performance. As software applications get ever more complex and intertwined and with the large number of different platforms and devices that need to get tested, it is more important than ever to have a robust testing methodology for making sure that software products/systems being developed have been fully tested to make sure they meet their specified requirements and can successfully operate in all the anticipated environments with the required usability and security. Testing can typically be broken down between functional and non-functional testing. Functional testing involves testing the application against the business requirements. It incorporates all test types designed to guarantee each part of a piece of software behaves as expected by using uses cases provided by the design team or business analyst. These testing methods are usually conducted in order and include:

- Unit testing
- Integration testing
- System testing
- Acceptance testing

Non-functional testing methods incorporate all test types focused on the operational aspects of a piece of software. These include:

- Performance testing
- Security testing
- Usability testing
- Compatibility testing

The key to releasing high quality software that can be easily adopted by your end users is to build a robust testing framework that implements both functional and non-functional software testing methodologies.

Unit Testing

The Unit testing part of a testing methodology is the testing of individual software modules or components that make up an application or system. These tests are usually written by the developers of the module and in a test-driven-development methodology (such as Agile, Scrum or XP) they are actually written before the module is created as part of the specification.

Integration Testing

The Integration testing part of a testing methodology is the testing of the different modules/components that have been successfully unit tested when integrated together to perform specific tasks and activities. This testing is usually done with a combination of automated functional tests and manual testing depending on how easy it is to create automated tests for specific integrated components.

System Testing

The system testing part of a testing methodology involves testing the entire system for errors and bugs. This test is carried out by interfacing the hardware and software components of the entire system (that have been previously unit tested and integration tested), and then testing it as a whole. This testing is listed under the black-box testing method, where the software is checked for user-expected working conditions as well as potential exception and edge conditions.

Acceptance Testing

The acceptance testing part of a testing methodology is the final phase of functional software testing and involves making sure that all the product/project requirements have been met and that the end-users and customers have tested the system to make sure it operates as expected and meets all their defined requirements

Performance Testing

Performance testing is a testing measure that evaluates the speed, responsiveness and stability of a computer, network, software program or device under a workload. Organizations will run performance tests in order to identify performance-related bottlenecks.

Security Testing

Security testing is a process intended to reveal flaws in the security mechanisms of an information system that protect data and maintain functionality as intended.

Due to the logical limitations of security testing, passing the security testing process is not an indication that no flaws exist or that the system adequately satisfies the security requirements.

Usability Testing

Usability testing is a way to see how easy to use something is by testing it with real users. Users are asked to complete tasks, typically while they are being observed by a researcher, to see where they encounter problems and experience confusion. If more people encounter similar problems, recommendations will be made to overcome these usability issues. Usability testing is a method used to evaluate how easy a website is to use. The tests take place with real users to measure how ‘usable’ or ‘intuitive’ a website is and how easy it is for users to reach their goals.

Compatibility Testing

Compatibility testing is software testing which comes under the non-functional testing category, and it is performed on an application to check its compatibility (running capability) on different platform/environments. This testing is done only when the application becomes stable. Means simply this compatibility test aims to check the developed software application functionality on various software, hardware platforms, network and browser etc. This compatibility testing is very important in product production and implementation point of view as it is performed to avoid future issues regarding compatibility.

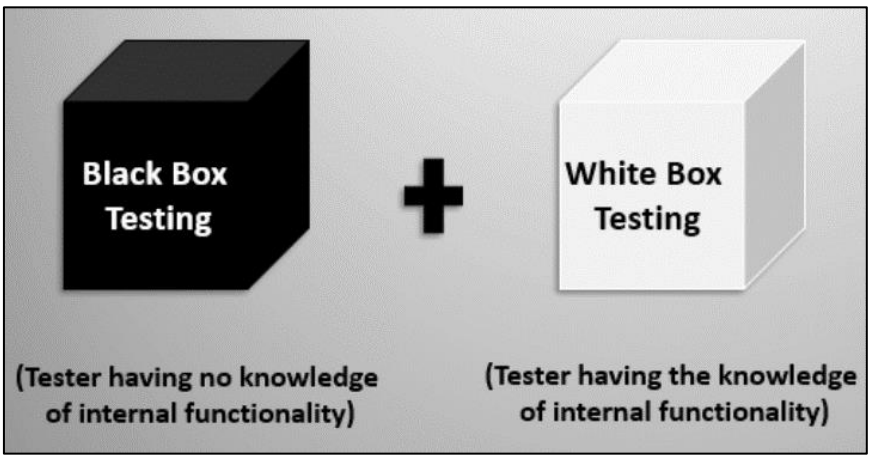
4.2 Testing Types

Black Box Testing

In Black-box testing, a tester doesn't have any information about the internal working of the software system. Black box testing is a high level of testing that focuses on the behavior of the software. It involves testing from an external or end-user perspective. Black box testing can be applied to virtually every level of software testing: unit, integration, system, and acceptance.

White Box Testing

White-box testing is a testing technique which checks the internal functioning of the system. In this method, testing is based on coverage of code statements, branches, paths or conditions. White-Box testing is considered as low-level testing. It is also called glass box, transparent box, clear box or code base testing. The white-box Testing method assumes that the path of the logic in a unit or program is known.



Key Difference Between Black Box Testing and White Box Testing

- In Black Box, testing is done without the knowledge of the internal structure of program or application whereas in White Box, testing is done with knowledge of the internal structure of program.
- When we compare Blackbox and Whitebox testing, Black Box test doesn't require programming knowledge whereas the White Box test requires programming knowledge.
- Black Box testing has the main goal to test the behavior of the software whereas White Box testing has the main goal to test the internal operation of the system.
- Comparing White box testing and Black box testing, Black Box testing is focused on external or end-user perspective whereas White Box testing is focused on code structure, conditions, paths and branches.
- Black Box test provides low granularity reports whereas the White Box test provides high granularity reports.
- Comparing Black box testing vs White box testing, Black Box testing is a not time-consuming process whereas White Box testing is a time-consuming process.

4.3 Test Cases

A **TEST CASE** is a set of actions executed to verify a particular feature or functionality of your software application. A Test Case contains test steps, test data, precondition, postcondition developed for specific test scenario to verify any requirement.

Test Case 1

Objective: Gadget Filtration

Input: Gadget brand, RAM, ROM

Output: List of gadgets that satisfies user desire

Pre-Requisite	Description	Steps	Expected Result	Actual Result	Status
User located to desired gadget tab	Basic data filtration, to produce resultant gadgets	1. Locate to desired tab 2. Select filters of choice	Gadgets display with respect to user choice	Gadgets display with respect to user choice	**Pass**
User located to desired gadget tab	Advance data filtration, to recommend similar gadgets	1. Locate to desired tab 2. Select filters of choice	When user combination unavailable, display results through content-based recommendation system	When user combination unavailable, display results through content-based recommendation system	**Pass**

Test Case 2

Objective: Prediction Form

Input: Gadget type, usage type, speed, photograph type, budget

Output: List of recommended gadgets that satisfies user desire

Pre-Requisite	Description	Steps	Expected Result	Actual Result	Status
User located to AI form prediction	User locates themselves to prediction form	1. Locate to AI form 2. Fill out the form 3. Resultant gadget displays	Random forest algorithm predicts user type and recommend gadget based on user type	Random forest algorithm predicts user type and recommend gadget based on user type	**Pass**

Test Case 3

Objective: Feedback Form

Input: User satisfaction, website known from, experience, recommendation

Output: Store data into database

Pre-Requisite	Description	Steps	Expected Result	Actual Result	Status
User located to Feedback form	User locates themselves to feedback form	1. Locate to feedback form 2. Fill out the form	Redirect page appears, user redirected to homepage	Redirect page appears, user redirected to homepage	**Pass**

Test Case 4

Objective: Analytics

Input: User prediction form and feedback form

Output: Google Datastudio Analytic

Pre-Requisite	Description	Steps	Expected Result	Actual Result	Status
Data stored into PostgreSQL from prediction form	Data has to be extracted and stored into PostgreSQL from prediction form	1. User fills in prediction form 2. Once submitted data stored into PostgreSQL "predictiondatabase"	User data stored into responsible database	User data stored into responsible database	**Pass**
Data stored into PostgreSQL from feedback form	Data has to be extracted and stored into PostgreSQL from feedback form	1. User fills in prediction form 2. Once submitted data stored into PostgreSQL "feedbackdatabase"	User data stored into responsible database	User data stored into responsible database	**Pass**
Data reflects into "Google Datastudio"	Datastudio visualization displayed in real time by connection with PostgreSQL table	1. Datastudio connection – AWS RDS – PostgreSQL 2. Visualizations displayed 3. Visualizations embedded through <iframe>	Visualizations displayed through google datastudio into website	Visualizations displayed through google datastudio into website	**Pass**

4.4 Test Reports

4.4.1 Testing & Result

Entire website was tested with several testing methodologies and it is working accurately. The website was tested as team effort from the developer as well as the testing team.

CHAPTER 5

SYSTEM

IMPLEMENTATION

5.1. Hardware required at Client side

- Processor: Intel core i5 or above
- Hard Disk: 10 GB or above.
- RAM: 4 GB or above.

5.2. Software required at Client side

Software

- Operating System: Windows
- Deployment server: Flask
- Front End: HTML, CSS, JavaScript
- Back End: PostgreSQL
-

Tools

- Database Server: Amazon Relational Database (Amazon RDS)
- Web Crawler: Beautiful Soup
- Data Visualization: Google Data Studio
- Data Prediction: Scikit-Learn - Random Forest (Machine Learning)

5.3 Software required at Client side

Acceptance Testing

It is a formal testing according to user needs, requirements and business processes conducted to determine whether a system satisfies the acceptance criteria or not and to enable the users, customers or other authorized entities to determine whether to accept the system or not.

Types of Acceptance Testing

- User Acceptance Testing (UAT)
- Business Acceptance Testing (BAT)
- Contract Acceptance Testing (CAT)
- Regulations Acceptance Testing (RAT)
- Operational Acceptance Testing (OAT)
- Alpha Testing
- Beta Testing

User Acceptance Testing (UAT)

User acceptance testing is used to determine whether the product is working for the user correctly. Specific requirements which are quite often used by the customers are primarily picked for the testing purpose.

Business Acceptance Testing (BAT):

BAT is used to determine whether the product meets the business goals and purposes or not. BAT mainly focuses on business profits which are quite challenging due to the changing market conditions and new technologies so that the current implementation may have to being changed which result in extra budgets.

Contract Acceptance Testing (CAT)

CAT is a contract which specifies that once the product goes live, within a predetermined period, the acceptance test must be performed and it should pass all the acceptance use cases. Here is a contract termed as Service Level Agreement (SLA), which includes the terms where the payment will be made only if the Product services are in-line with all the requirements, which means the contract is fulfilled. Sometimes, this contract happens before the product goes live. There should be a well-defined contract in terms of the period of testing, areas of testing, conditions on issues encountered at later stages, payments, etc.

Regulations Acceptance Testing (RAT)

RAT is used to determine whether the product violates the rules and regulations that are defined by the government of the country where it is being released. This may be unintentional but will impact negatively on the business.

Generally, the product or application that is to be released in the market, has to go under RAT, as different countries or regions have different rules and regulations defined by its governing bodies. If any rules and regulations are violated for any

country then that country or the specific region then the product will not be released in that country or region. If the product is released even though there is a violation then only the vendors of the product will be directly responsible.

Operational Acceptance Testing (OAT)

OAT is used to determine the operational readiness of the product and is a non-functional testing. It mainly includes testing of recovery, compatibility, maintainability, reliability etc. OAT assures the stability of the product before it is released to the production.

Alpha Testing

Alpha testing is used to determine the product in the development testing environment by a specialized testers team usually called alpha testers.

Beta Testing

Beta testing is used to assess the product by exposing it to the real end-users, usually called beta testers in their environment. Feedback is collected from the users and the defects are fixed. Also, this helps in enhancing the product to give a rich user experience.

CHAPTER 6

SYSTEM MAINTAINANCE

AND EVALUATON

6.1 Software Maintenance

Software maintenance is a part of Software Development Life Cycle. Its main purpose is to modify and update software application after delivery to correct faults and to improve performance. Software is a model of the real world. When the real-world changes, the software requires alteration wherever possible.

Software maintenance is a vast activity which includes optimization, error correction, deletion of discarded features and enhancement of existing features. Since these changes are necessary, a mechanism must be created for estimation, controlling and making modifications. The essential part of software maintenance requires preparation of an accurate plan during the development cycle. Typically, maintenance takes up about 40-80% of the project cost, usually closer to the higher pole. Hence, a focus on maintenance definitely helps keep costs down.

Software Maintenance Processes

- The SM process includes a maintenance plan which contains software preparation, problem identification and find out about product configuration management.
- The problem analysis process includes checking validity, examining it and coming up with a solution and finally getting all the required support to apply for modification.

- The process acceptance by confirming the changes with the individual who raised the request.
- The platform migration process, which is used if software is needed to be ported to another platform without any change in functionality.

Some software points that affect maintenance cost include

- Structure of Software Program
- Programming Language
- Dependence on external environment
- Staff reliability and availability

Types of maintenance

In a software lifetime, type of maintenance may vary based on its nature. It may be just a routine maintenance tasks as some bug discovered by some user or it may be a large event in itself based on maintenance size or nature. Following are some types of maintenance based on their characteristics:

Corrective Maintenance

This includes modifications and updating done in order to correct or fix problems, which are either discovered by user or concluded by user error reports.

Adaptive Maintenance

This includes modifications and updating applied to keep the software product

up-to date and tuned to the ever-changing world of technology and business environment.

Perfective Maintenance

This includes modifications and updates done in order to keep the software usable over long period of time. It includes new features, new user requirements for refining the software and improve its reliability and performance.

Preventive Maintenance

This includes modifications and updating to prevent future problems of the software. It aims to attend problems, which are not significant at this moment but may cause serious issues in future.

6.2 Software Evaluation

A software evaluation is a type of assessment that seeks to determine if software or a combination of software programs is the best possible fit for the needs of a given client. The idea is to look closely at the resources and tools provided by the software that is either currently in use or is being examined as a possible addition to programs already in use by that client. Based on a prepared list of criteria along with some practical experimentation, a software evaluation makes it possible to determine if the products would be helpful to the client or if some other combination of software products would serve to better advantage.

Process for system evaluation

Ensure that necessary, adequate data are acquired and analyzed to determine and assess the advisability, adequacy and effectiveness of the management system in the areas of quality, environment and occupational health and safety, including the Policy and Objectives, as well as identifying and establishing guidelines and actions for improvement.

Integration

There is a high degree of coincidence between what is required in the ISO 9001: and what is required in the ISO 14001: and the OHSAS 18001: with regard to system evaluation (analysis and review) by management. In all cases, the

requirements point out the need to analyze the data to obtain relevant information and the use of this information by Management for review of the management system. Therefore, it may be said that, regardless of the area, this process always has the same structure. Another aspect that promotes integration in this process is precisely the fact that the Management, as shown by its commitment, must assume the responsibility for system review. It is a common practice to hold periodic meetings for this activity. Seeking integration of the quality, environmental and occupational health and safety management systems implies that these scheduled meetings must permit joint, simultaneous analysis and review. This simultaneous analysis and review of the three areas further promotes the understanding of the interrelationships between them so that decisions and guidelines taken are more compensated and balanced.

Indicators

The indicators that are proposed are the same regardless of the area in which they are to be applied, and can be calculated for the three systems together or independently.

CHAPTER 7

USER/OPERATIONAL

MANUAL

7.1 Security Aspects, Access Rights

Security is freedom from, or resilience against, potential harm (or other unwanted coercive change) caused by others. Beneficiaries (technically referents) of security may be of persons and social groups, objects and institutions, ecosystems or any other entity or phenomenon vulnerable to unwanted change.

Z'elle Security Goals

- **Privacy** - Information within our infrastructure and systems will only be accessible by authorized users
- **Integrity** - Data and information within our infrastructure cannot be tampered with by any unauthorized user
- **Data Protection** - Data within the systems cannot be harmed, deleted or destroyed
- **Identification and Authentication** - Ensures that any user of the system is who he claims to be and eliminates chances of impersonation
- **Network Service Protection** - Ensures that networking equipment is protected from malicious hacking attempts or attacks that threaten uptime

7.2. Back Up

Amazon Simple Storage Service (Amazon S3) – (Future Enchancement)

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. This means customers of all sizes and industries can use it to store and protect any amount of data for a range of use cases, such as data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics. Amazon S3 provides easy-to-use management features so you can organize your data and configure finely-tuned access controls to meet your specific business, organizational, and compliance requirements. Amazon S3 is designed for 99.999% (11 9's) of durability, and stores data for millions of applications for companies all around the world.

CHAPTER 8

FUTURE

ENHANCEMENTS

User Profile Facilities

Current version of Z'elle stands a blog, such that users can visit the website and find their desired gadget or be recommended by Z'elle. In the near future a user profile will be maintained to collect information associated with a user. Information used to identify an individual, such as their name, age, portrait photograph and individual characteristics such as knowledge or expertise. This will enable the website to offer much personalized information to the users.

Gadget Purchase Options and Comparison

Once a user finds a gadget useful the user may store the product in a list where they can further compare the gadget with other gadgets of their choice and then make a purchase of the most desired gadget.

Data Automation through AWS Lambda

In the current version of this website bots are used to extract the data from the internet. These bots are run every time the application is restarted. AWS Lambda is an event-driven, serverless computing platform provided by Amazon as a part of Amazon Web Services. It is a computing service that runs code in response to events and automatically manages the computing resources required by that code.

Data Storage in AWS S3

As the website grows, so will the data from web scraping and user data. Storing the data within the AWS RDS will not be a reasonable solution, thus data can be stored in AWS S3 and cached in AWS RDS. Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. This means customers of all sizes and industries can use it to store and protect any amount of data for a range of use cases, such as data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics.

CHAPTER 9

LIMITATIONS

Limitations

- Z’elle runs on localhost thus not available to a wider audience
- Online purchase is not possible
- This website is automated with several bots that take long computational time and space
- Data is stored in AWS RDS – PostgreSQL, which may not be able to store massive data with an increase in website data
- User interface is not very aesthetic
- Google Datastudio has limited features, thus Z’elle is unable to optimize the data in the most visual ways as offered by paid software’s such as tableau and power BI

CHAPTER 10

CONCLUSION

CONCLUSION

The project training undertaken by me - Ginelle D'souza named Z'elle – TechNoStress, helped me to gain practical experience but also to adapt the organizational environments and procedures. This system gives satisfactory outputs. This website has been tested for all possible exceptions keeping in mind, the primary requirement of this project. This experience has enriched my knowledge of developing software, machine learning algorithms and data analytics. It has proved to be a stepping stone in my career. It is said that no system is perfect; every system has some limitations and flaws. Taking this into consideration, the Z'elle – TechNoStress has been carried out successfully.

CHAPTER 11

BIBLIOGRAPHY

BIBLIOGRAPHY

Reference Websites

- http://www.w3schools.com
- http://en.wikipedia.org
- https://www.geeksforgeeks.org
- https://aws.amazon.com
- https://medium.com
- https://machinelearningmastery.com
- https://scikit-learn.org
- https://stackoverflow.com
- https://www.google.com
- https://github.com