

Team ID	PNT2022TMID42787
Project Name	News Tracker Application

A PROJECT REPORT

Submitted by

GANAPATHY S (711619106007)

KARTHICK SARAN K (711619106014)

GOPINATH M (711619106009)

SNEHA S (711619106027)

SAMSON DURAI M (711619106023)

In partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING



KATHIR COLLEGE OF ENGINEERING

“WISDOM TREE”, NEELAMBUR, COIMBATORE – 641 062

ANNA UNIVERSITY: CHENNAI – 600 025

ABSTRACT

As world's technology is rapidly growing we has fast connection and network to instantly connect to other person. Day to day use in mobile, tablets and laptop is increasing, most of the people already have this facilities. In this fast and information oriented world we need to stay updated with every incidents and news too. This News app is android mobile application where user have access to latest news from 120+ newspapers from 50+ countries. The main focus of this application is to connect news articles from all around the world and deliver it to user as fast as possible in best visualize way.

As news is increasingly accessed on smartphones and tablets, the need for personalizing news app interactions is apparent. We report a series of three studies addressing key issues in the development of adaptive news app interfaces. We first surveyed users' news reading preferences and behavior; analysis revealed three primary types of reader. We then implemented and deployed an Android news app that logs users' interactions with the app. We used the logs to train a classifier and showed that it is able to reliably recognize a user according to their reader type. Finally we evaluated alternative, adaptive user interfaces for each reader type. The evaluation demonstrates the differential benefit of the adaptation for different users of the news app and the feasibility of adaptive interfaces for news apps.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	
	LIST OF FIGURES	
	LIST OF ABBREVIATIONS	
1	INTRODUCTION	
	1.1 Project overview	
	1.2 Purpose	
2	LITERATURE SURVEY	
	2.1 Existing Problem	
	2.2 References	
	2.3 Problem Statement Definition	
3	IDEATION AND PROPOSED SOLUTION	
	3.1 Empathy Map Canvas	
	3.2 Ideation & Brainstorming	
	3.3 Proposed Solution	
	3.4 Problem Solution Fit	
4	REQUIREMENT ANALYSIS	
	4.1 Functional requirement	
	4.2 Non – Functional requirement	

5	PROJECT DESIGN
5.1	Data Flow Diagrams
5.2	Solution & Technical Architecture
5.3	User Stories
6	PROJECT PLANNING & SCHEDULING
6.1	Sprint Planning & Estimation
6.2	Sprint Delivery Schedule
6.3	Reports from JIRA
7	CODING & SOLUTIONING
7.1	Feature 1
7.2	Feature 2
7.3	Database Schema
8	TESTING
8.1	Test Cases
8.2	User Acceptance Testing
9	RESULTS
9.1	Performance Metrics
10	ADVANTAGES & DISADVANTAGES
11	CONCLUSION
12	FUTURE SCOPE
13	APPENDIX

Source Code,

GitHub & Project demo link

CHAPTER -1

TITLE: NEWS TRACKER APPLICATION

1.INTRODUCTION

The **News** app allows users to browse through news headlines for different categories such as Top Stories, General, Business, Sports, Technology, Entertainment, and so on. Furthermore, the app lets user glance through the news and weather conditions. The app makes use of Integration and Orchestration services of Quantum Fabric to fetch data from third-party APIs, and then populates the data to the front end of the app.

Personalization of news access clearly needs to extend beyond ‘what’ content users access to ‘how’ they access it, as evident in the abundance of mobile news apps offering personalization features.

For example,

News Break.com – Breaking News allows users to select news topics to follow and then provides 300-character summaries of relevant stories along with links to the original sources. Another example is Newsbeat, again an aggregator but one that creates ‘personalized radio news bulletins. Users select their preferred text news sources from which stories are pulled each day, summaries created, then news podcasts created using text-to-voice technology. A third example is Flipboard, which uses the metaphor of a ‘personal magazine’ to present articles from conventional news providers as well as social media updates, and RSS feeds. Users curate and share their own mini-magazines within the app, drawing in stories on their preferred topics.

The personalization of news app interaction in these examples is achieved through making the interface adaptable. Adaptive news interfaces that ‘automatically’ adapt to the way the user reads the news in particular contexts are not found, other than in re-ordering menus of headlines to take account of previous reading choices

1.1 Project Overview:

In the current scenario, there is no single platform (in application) present right now which provides security information, E- sport information, Science and Technology information, etc. in one place Cybersecurity users have to visit different websites to gather the news related to the cyber world. Many people do not have the time to visit different websites to gather information. Ultimately this would be a waste of time and effort. Visiting different websites, the user might get the redundancy in the information.

Mobile app ecosystems are transforming patterns of news consumption. Until quite recently, reading the news was a niche use for smartphones, mostly for when users were 'on the go'; now however, two in every three users of mobile devices in the US regularly access news and as many as one in five read in-depth news articles daily; a similar picture is found in the UK. This growth in mobile news access continues the migration of news consumers to the Internet

1.2 Purpose:

The purpose is to develop an android application, which will eliminate the problems faced in the current scenario. This application will provide all the information and news related to the cybersecurity, E-sport, Science, and Technology or that are in trend tone place. So, it will save time and efforts of the users by making it more efficient. Using, this application will terminate the possibility of information redundancy.

Adaptive personalization relies on constructing and exploiting an individual user profile to deliver a tailored version of the user interface. A user profile is a model of the user that the system learns through interaction with the user. The construction of a user profile can be based on explicit or implicit information gathering approaches. The former consists of information provided directly by the user such as by forms and questionnaires; although data captured this way will have a greater reliability, the disruption to the user can be significant.

CHAPTER -2

2.LITERATURE SURVEY:

A literature review is a piece of discursive prose to find objectivity and facts in the current information. We follow a thorough iterative assessment process to distil information from credible and highly impactful sources.

2.1 Existing Problem:

The mobile ecosystem has various crisis and news application is one of them. Mobile news access perfectly complements the continuously updating, 24-hour nature of digital news services. But if users are now never out of range of the news, they need more than ever for that access to be adaptive and personalized. Personalized news services are already able to help people find news that is relevant to them, to recommend the right news to the right users, and to help users keep abreast of news by aggregation over multiple sources.

The problems still exist in the news application are

- Spammed with Notifications.
- Fake news getting shared.
- Increase in Number of irrelevant news.
- Too many apps used by a single user.

This adaptivity is achieved through several methods including: news content personalization by pushing filtered articles predicted to match the user's interests; adaptive news browsing by changing the order of news categories; contextual news access by offering users access to additional information related to the news they are reading; and news aggregation, by automatically identifying main news topics emerging from multiple sources. This previous work on adaptivity in digital news access has focused on recommendation of news content. But, adaptation of the way people interact with news services has not been investigated.

2.2 References:

NEWS TRACKER APPLICATION

S.NO	PAPER	AUTHOR	YEAR	METHOD AND ALGORITHM
1.	Detection and Tracking in News Articles	Sagar Patel Sanket Suthar Sandip Patel Neha Patel	MARCH 2017	We have presented an idea in this paper for detecting and tracking topics from news articles. Topic detection and tracking are used in text mining process. From data which are unstructured in text mining we extracts previously unknown and useful information. The main purpose of this paper is to identify and follow tasks occurred in different news sources. We are going to use agglomerative clustering based on average linkage for detecting the topics, calculate the similarity of topics using cosine similarity and KNN classifier for tracking the topics Detecting , tracking ,article ,text mining, extract, unstructured
2.	Exploring mobile news reading interactions for news app personalisation	Marios Constantinides John Dowell David Johnson Sylvain Malacria	AUGUST 2015	As news is increasingly accessed on smartphones and tablets, the need for personalising news app interactions is apparent. We report a series of three studies addressing key issues in the development of adaptive news app interfaces. We first surveyed users' news reading preferences and behaviours; analysis revealed three primary types of reader. We then implemented and deployed an Android news app that logs users' interactions with the app. We used the logs to train a classifier and showed that it is able to reliably

				recognise a user according to their reader type. Finally we evaluated alternative, adaptive user interfaces for each reader type. The evaluation demonstrates the differential benefit of the adaptation for different users of the news app and the feasibility of adaptive interfaces for news apps.
3.	A simple location-tracking app for psychological research	Kristoffer Geyer David A. Ellis Lukasz Piwek	NOVEMBER 2018	Location data gathered from a variety of sources are particularly valuable when it comes to understanding individuals and groups. However, much of this work has relied on participants' active engagement in regularly reporting their location. More recently, smartphones have been used to assist with this process, but although commercial smartphone applications are available, these are often expensive and are not designed with researchers in mind. To overcome these and other related issues, we have developed a freely available Android application that logs location accurately, stores the data securely, and ensures that participants can provide consent or withdraw from a study at any time. Further recommendations and R code are provided in order to assist with subsequent data analysis.
4.	Infant Feeding Tracker Applications: Cross-Sectional Analysis of Use	Lauren M Dinour	MAY 2022	Objective Examine the extent to which postpartum patients use infant feeding tracker applications (apps), characteristics of app users, and app features most used and desired. Design Cross-sectional

				<p>survey. Setting An obstetrics/gynecology practice in Northern New Jersey in 2019.</p> <p>Participants One hundred twenty-six patients aged ≥ 18 years recruited during their 6-week postpartum visit.</p> <p>Main Outcome Measures</p> <p>Self-reported sociodemographics, infant feeding behaviors, and health app use. Respondents were grouped by self-reported use of an infant feeding tracker app.</p> <p>Analysis Frequencies were calculated for descriptive analysis, and comparisons were made between user groups. Fisher's exact tests of independence were used to analyze categorical data. Mann-Whitney U tests were employed for continuous variables (significance at $P < 0.002$).</p> <p>Results Fifty-seven percent of respondents reported using an app to track infant feeding. Compared with nonusers, users were more likely to have an infant who was ever breastfed ($P = 0.001$; Cramer's $V = 0.30$) and exclusively breastfed ($P < 0.001$; Cramer's $V = 0.44$). Users also used significantly more health apps than nonusers ($P < 0.001$).</p> <p>Most respondents used the app to track infant feeding, diapering, and sleep.</p> <p>Conclusions and Implications Given their frequency of use—particularly among those who breastfeed—infant feeding tracker apps have the potential to support parents in meeting their infant feeding goals.</p>
--	--	--	--	--

5.	Following the Fed with a News Tracker	Michael William Mccracken	JANUARY 2012	How should one conclude whether the data have come in stronger, weaker, or as expected?>
6.	New tracker models of dark energy	Satadru Bag Swagat Saurav Mishra Varun Sahni	SEPTEMBER 2017	. We describe a new class of dark energy (DE) models which behave like cosmological trackers at early times. These models are based on the α -attractor set of potentials, originally discussed in the context of inflation. The new models allow the current acceleration of the universe to be reached from a wide class of initial conditions. Prominent examples of this class of models are the potentials $\coth \varphi$ and $\cosh \varphi$. A remarkable feature of this new class of models is that they lead to large enough negative values of the equation of state at the present epoch, consistent with the observations of accelerated expansion of the universe, from a very large initial basin of attraction. They therefore avoid the fine tuning problem which afflicts many models of DE.

7.	Architecture for Lost Mobile Tracker Application	Jude Martinez Nelson Widjaya	JUNE 2012	<p>The aim of this paper is to analyze existing mobile application's architecture and design a new one based on it, in addition, it seeks to implement the application prototype as a proof of concept of the designed architecture. The main benefit of this research is to provide a base architecture for java mobile developers in order to develop a tracking application for lost mobile phones. The observations gathered on existing mobile applications show that most of the mobile applications in tracking lost phones are focused on three characteristics, which are client to server communication, SIM card data retrieval, and mobile application activation event. The result of this research is the application prototype that tracks lost mobile phones. The application prototype implements the architecture analyzed from the characteristics of existing mobile applications which are used to track lost mobile phones</p>
8.	Financial indicators tracker application (FIT)	Khairal Shafee Kalid Md Akhir Mohd Sharif	JUNE 2012	<p>The Internet has become a source of vast range of financial information from financial institutions. The existing financial data provider such as Google finance, Yahoo finance, Bloomberg and Reuters are those leaders in providing financial data for all company in the world. Currently, these providers only provide financial data without any financial analysis. In order to do a financial analysis, users would have to browse the web sites of these providers and calculate manually the financial ratios.</p>

2.3 Problem Statement Definition:

- News organizations whose mobile apps only provide users with their articles or videos are missing a big opportunity.
- An application, by definition, should be applied to perform a task, to solve a problem. Most news doesn't do that.
- Rather than just feed readers recent stories you wrote about their problems, apps can provide tools and data that enable users to actually solve their problems. When you solve problems, you get more loyal users.



Why this?

Among all the choices given to us, why did we choose this? Logically speaking, we, as a team, felt that the industry has a wide audience range from which we could benefit. And well melodramatically speaking, as rare users of the news app, we wondered about all the problems users were facing in this area. We felt that news apps are used by a majority of the population but still might be a neglected space.

I personally felt that since almost everyone has a news app these days, this topic would have great research potential and we would actually be covering real problems faced by users

Problem boundaries:

- Fake news and information
- No feedback and comment section
- No Sharing Option
- Instant Notifications

Features of our Product:

1. **Personalization Page** First-time users will get an option to answer a few questions based on which the app customizes the Feed of the user. The users can also edit their customization over the content whenever they want.
2. **Discussion Rooms** Users will have an option to join category-based discussion rooms via Chat Room or Audio Room. So that they can participate and have a discussion or debate over a piece of certain news or a news category.
3. **Paid Premium Plans** For the users who want to stop the advertisements that come up while using the apps, they can upgrade to a Paid Premium Plan.

However, if users are not

willing to pay, they can also get upgraded if they: — Invite their friends onto the platform, or — Rate the product on the play store.

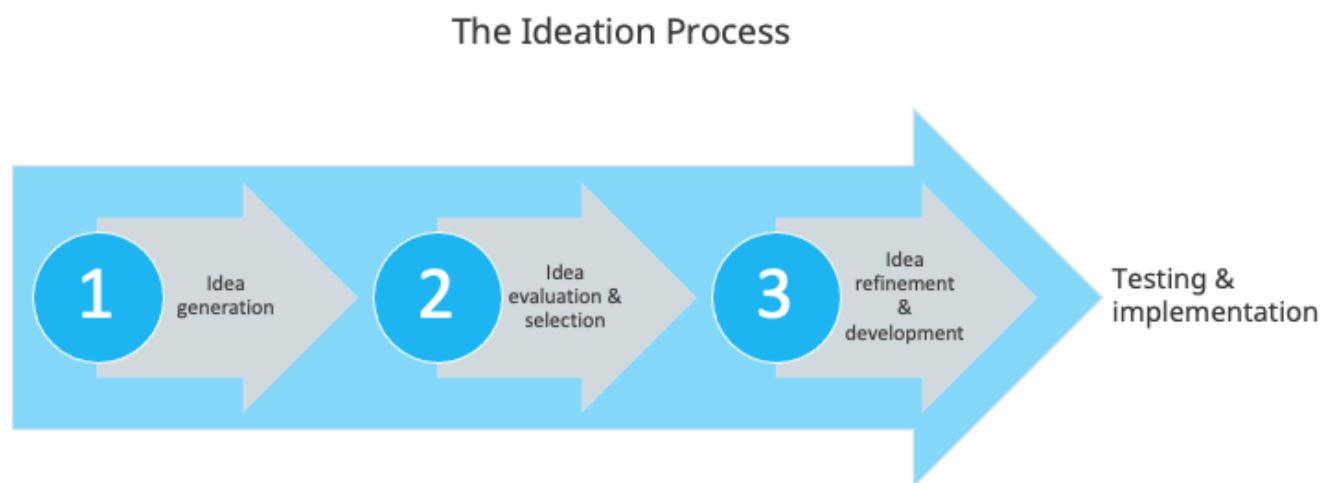
4. **Personalized Notifications** — Users can set a timer on what time they want to receive the notifications from the app. — They can prioritize the notifications by choosing the categories. — They can also decide the number of notifications they want to receive per day.

CHAPTER-3

3. IDEATION & PROPOSED SOLUTION:

Ideation is **the creative process of generating new ideas**, which can be accomplished through a variety of ideation techniques, such as brainstorming and prototyping. If done right, ideation is what helps founders and executives determine the right problem to solve and how to solve it

THREE STAGES OF IDEATION:



Why Ideation is Important?

- It can increase your innovation opportunities
- Ideation can bring together different perspectives and ways of thinking
- Develop and refine those into better ones.

3.1 Empathy Map Canvas:

An empathy map is a widely-used visualization tool within the field of UX and HCI practice. In relation to empathetic design, the primary purpose of an empathy map is to bridge the understanding of the end user.

An empathy map is **a collaborative tool teams can use to gain a deeper insight into their customers**. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community.

Why is empathy map important?

An empathy map helps to map what a design team knows about the potential audience. This tool helps to understand the reason behind some actions a user takes deeply. This tool helps build Empathy towards users and helps design teams shift focus from the product to the users who are going to use the product

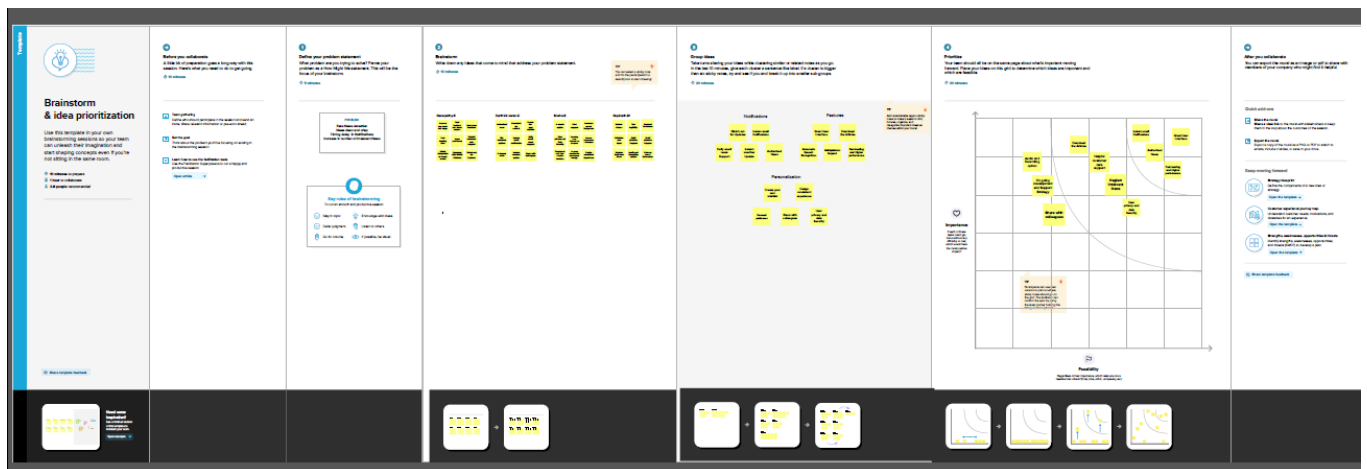


3.2 IDEATION & BRAINSTROMING:

What is ideation in problem solving?



Ideation is **the creative process of generating new ideas**, which can be accomplished through a variety of ideation techniques, such as brainstorming and prototyping. If done right, ideation is what helps founders and executives determine the right problem to solve and how to solve it.



PROBLEM:

Fake News Detection
News clean and Crisp
Time Delay in Notifications
Increase in Number of Irrelevant News

BRAINSTORM IDEA POINTS:

- Improve app speed and usage
- Good feedback and Review session
- Great User Interface
- Fast Loading Time
- Higher Performance
- Helpful Customer care Support
- Quick adaption to User Needs
- Compact with Mobile Platform
- Watch Out for Updates
- Create the Own Wishlist
- Authorized news
- Like Comments options
- Instant E-mail notification
- Daily E-mail news support
- Instant weather updates
- Category news
 - Share with colleagues
- A Search Option
- User privacy and Data Security
- Providing more Flexibility for everyone
- Battery Preservation Techniques
- Responsive and User Friendly Interface
- Ongoing Development and Support Strategy
- Design Consistent Experience
- Reduce Clutter
- Text legibility
- Personal preference
- Minimize User Input
- Multi Platform Support
- Neglect Irrelevant News
- Minimum cost of Subscription

3.3 Proposed Solution:

Proposed Solution means the combination of software, hardware, other products or equipment, and any and all services (including any installation, implementation, training, maintenance and support services) necessary to implement the solution described by Vendor in its Proposal.

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	<ul style="list-style-type: none">• Fake news detection• News clean and crisp• Time delay in Notifications• Increase in number of irrelevant news
2.	Idea / Solution description	<ul style="list-style-type: none">• Improve app speed and usage• Automatic speech recognition• User privacy and data security• Create own Wishlist• Authorized news• Compact with mobile platform
3.	Novelty / Uniqueness	<ul style="list-style-type: none">• Fast loading and higher performance• Category news• Easy access and able to share with the colleagues• Helpful customer care support
4.	Social Impact / Customer Satisfaction	<ul style="list-style-type: none">• Daily email news report• Providing more flexibility for everyone• On going development and support strategy• Neglect irrelevant news
5.	Business Model (Revenue Model)	<ul style="list-style-type: none">• Affordable cost of subscription• Multiplatform Support• Design consistent experience• Minimization of user input and work progress
6.	Scalability of the Solution	In scale of 10 : <ul style="list-style-type: none">• As a product - 8.5• In point of user- 8• As a business- 9• As a developer-9.2

3.4 Problem Solution Fit:

- Problem-solution fit is a term used to describe the point validating that the base problem resulting in a business idea really exists and the proposed solution actually solves that problem. Validate that the problem exists: When you validate your problem hypothesis using real-world data and feedback. The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem.

PROBLEM -SOLUTION FIT				
Define CS, fit into	1. CUSTOMER SEGMENT(S) CS 1. Habitual Readers 2. Light Readers 3. Zombie Readers	6. CUSTOMER CONSTRAINTS CC 1. Minimum cost of subscription. 2. Amount of time spending for an news application. 3. Instant news updates and instant notifications. 4. Weather report and climatic changes according to the location and the users Convenience. 5. Review by the friends and family members and their way of suggestion.	5. AVAILABLE SOLUTIONS AS 1. Solid newspapers for reading with printed version of news updates . 2. Various news app can suggest new various and useful stuffs. 3. Readers can read at any time as their wish and willing or intrest to be.	Explore AS,
	2. JOBS-TO-BE-DONE / PROBLEMS J&P 1. Fake news detection 2. Instant updates and notifications 3. Time delay in news updates 4. Increase in number of irrelevant news 5. Do not know how to categorize the news and the type of news	9. PROBLEM ROOT CAUSE RC 1. Problem of gaining the trust and also the customers(users). 2. Due to raise in online news sharing platforms during the frequent improvement in the period of pandemic people lost the habit of reading and theirs concern about the knowledge of happenings in other places.	7. BEHAVIOUR BE e1. Add user friendly Interface where the customer can interact and find the various suggestions and customer care support to get the queries get cleared 2. Easy accesess of user interface and minimum field of input requirement 3. It should provide only publish the preferred And important suggested news also the Details provided must be precise enough.	Focus on J&P, tap into BE, understand
Identify strong TR & EM	3. TRIGGERS The users is unable to reach and fulfill the needs of Their customers due to which the news tracker app Is not getting improved and wants a solution. TR	10. YOUR SOLUTION 1. Improve app speed and usage 2. Automatic speech recognition User privacy and data security 3. Create own Wishlist 4. Authorized news 5. Compact with mobile platform SL	8. CHANNELS of BEHAVIOUR CH 1. Design consistent experience 2. Daily email news report 3. Providing more flexibility for everyone 3. On going development and support strategy 4. Affordable cost of Subscription 5. Multiplatform Support	Extract online & offline CH of BE
	4. EMOTIONS: BEFORE / AFTER The creators and developers of the application feels less Interactive with the customer as we are not getting Instant Notifications and updates upto the market and feels insecure and drowns to express and causes failures EM			

CHAPTER-4

4. REQUIREMENT ANALYSIS:

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem. Requirements analysis or requirements engineering is a process used to determine the needs and expectations of a new product. It involves frequent communication with the stakeholders and end-users of the product to define expectations, resolve conflicts, and document all the key requirements

Requirements analysis involves organizing requirements, modelling requirements and designs, validating information, identifying solutions that answer the business needs, and assessing the potential value gained by performing these tasks.

Once the business analyst gathers all of the necessary information through elicitation efforts the business analysts can start their work on **requirements analysis and design definition**. The analyzed, structured, and stakeholder-verified requirements lay a foundation for design definition and ultimately developing a solution that will bring value to the organization.



4.1 Functional Requirements:

Following are the functional requirements of the proposed solution:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through Linked in
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Account Sign up	User ID or Registered Phone Number Password
FR-4	User Interoperability	Requirement describes whether a software system is interoperable across different systems.
FR-5	User Technical Compliance	Compliance functional requirements validate that the developed system is compliant to Industrial standards
FR-6	User Requirement ID and Description	This attribute mentions the unique requirement id. Requirement Id is used in tracking the requirements in the database easily and also in mapping the requirements in code efficiently.

4.2 Non– Functional Requirements:

Following are the non-functional requirements of the proposed solution:

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The application should be user-friendly. Navigation should be simple. It should be easy to learn.
NFR-2	Security	The functional requirement describes the security aspect of software requirements
NFR-3	Reliability	It Should give the correct results consistently. The application performs the function that the user expected. The system prevents any unauthorized access and abuse.
NFR-4	Performance	Performance means system throughput under a given workload for a specific timeframe. Performance is validated by testing the scalability and the reliability of hardware, software and network
NFR-5	Availability	Availability is when your apps remain available and accessible without any interruption and serve their intended function seamlessly
NFR-6	Scalability	Scalability simply refers to the ability of an application or a system to handle a huge volume of workload or expand in response to an increased demand for database access, processing, networking, or system resources.

CHAPTER-5

5. PROJECT DESIGN:

Project design is an early phase of the project lifecycle where ideas, processes, resources, and deliverables are planned out. A project design comes before a project plan as it's a broad overview whereas a project plan includes more detailed information.

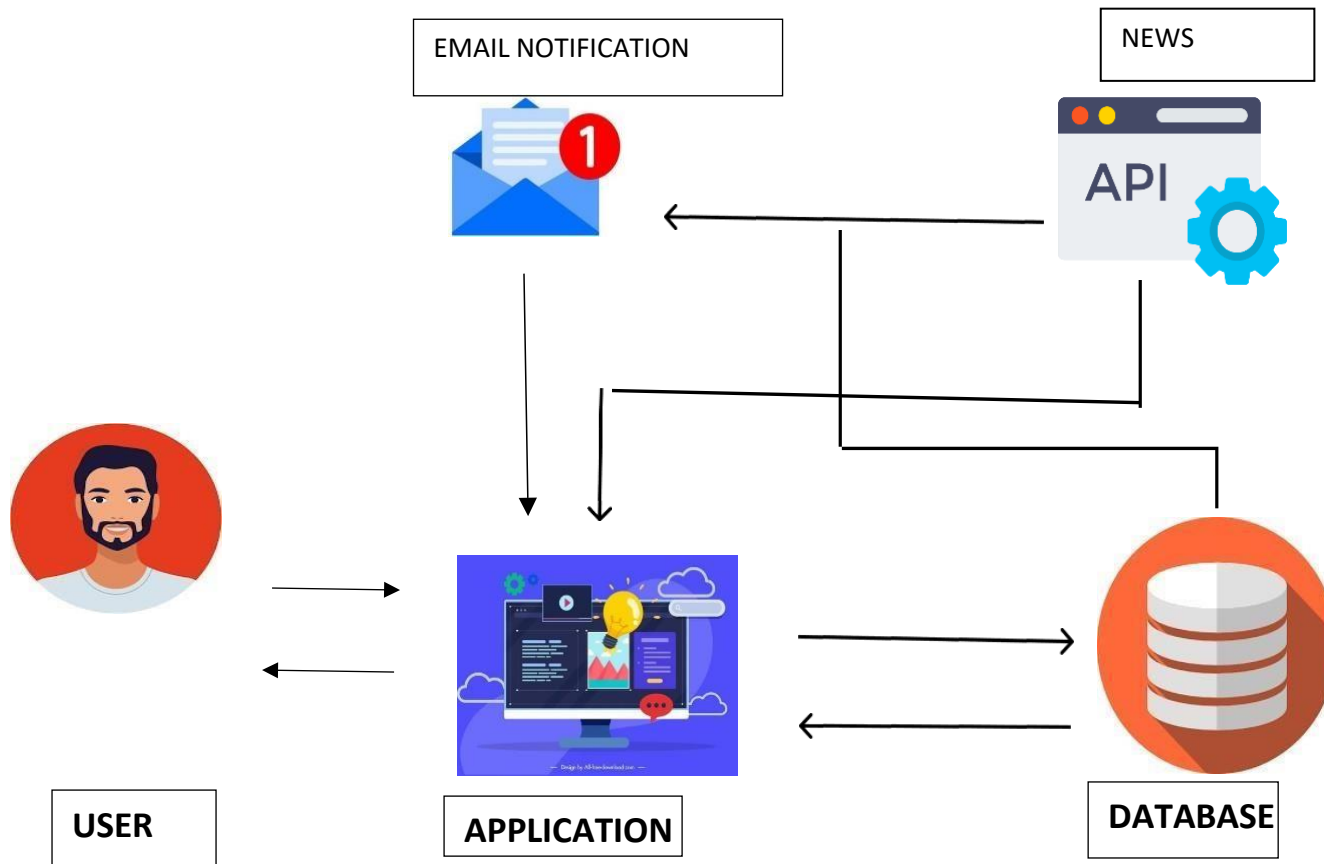
They help your team understand how to move through a project in the correct way. They help you avoid omitting important steps or items. They help you look more professional. They put the 'know how' in the business, instead of in employees.

5.1 Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually “say” things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That’s why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

Example: (Simplified)



step 1: Sign up for a Developer account.

Step 2: Accept the Distribution Agreement.

Step 3: Pay registration fee if user go for an advertisement free subscription.

Step 4: Complete your account details

Step 5: kindly provide with proper proof and authority.

Step 6: After receiving Email or messages get OTP for Verification Process.

Step 7: Registration Id / user Name and set up user Password/Security key. Then User can create an Wishlist and apply filters for their convenience.

Receive Instant updates and Proper Notifications.

5.2 SOLUTION & TECHNICAL ARCHITECTURE:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

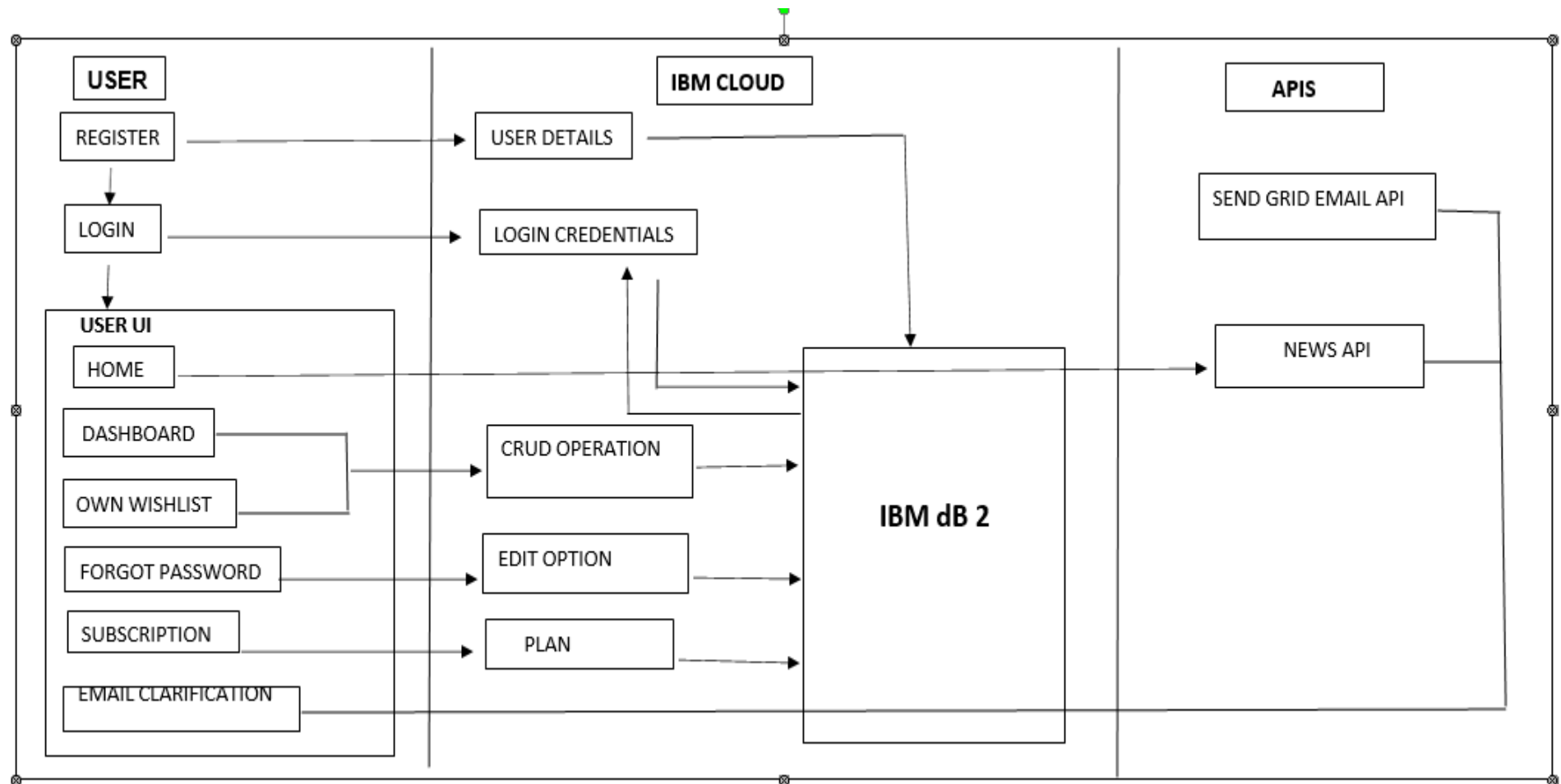


Table 1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web UI	HTML5, CSS3, Bootstrap3, Java Script
2.	Application Logic-1	Logic for a process in the application	Python Flask
3.	Application Logic-2	Logic for a process in the application	Send Grid Service
4.	Application Logic-3	Logic for a process in the application	News API
5.	Database	Data Type, Configurations etc.	MySQL
6.	Cloud Database	Database Service on Cloud	IBM DB2
7.	File Storage	File storage requirements	IBM Object Storage
8.	External API-1	Purpose of External API used in the application	News API
9.	External API-2	Purpose of External API used in the application	Google News API
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Docker, Kubernetes

Table-2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Bootstrap3, Flask
2.	Security Implementations	List all the security / access controls implemented,use of firewalls etc.	HTTPS, SSL
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Micro Service
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Load Balancer handled by IBM
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	IBM free tier

5.3 User Stories:

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
Customer (confirmation)	Confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
Customer (registration to different platforms)	Facebook support	USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
Customer (Gmail login)	Gmail Login (optional)	USN-4	As a user, I can register for the application through Gmail	I can access the application through my Gmail account	Medium	Sprint-1
User (Account Login)	Login	USN-5	As a user, I can log into the application by entering email & password	Easy to access the application with already registered details and passwords	High	Sprint-1
Application (front end)	Dashboard	USN-6	As User, can login to the application using their I'd and password	Two simple step Process	Medium	Sprint-1
Customer (Web user)	Web portal	USN-7	As User the link of the web portal of the application has to be created for sharing the Content	Web link creation and separate web portal for user	High	Sprint-1
Customer Care Executive	Customer Support	USN-8	As the user can expect customer care support 24/7 to resolve the problems .	Customer care and support	High	Sprint-1
Administrator	Admin page	USN-9	kindly provide with proper proof and authority. And admin gets once clarified.	Admin gets the complete details can able to provide user their needs and security.	High	Sprint-1
Designer type	Design language	USN-10	As in point of User expects some different design languages.	Design in the application gets some attention.	Medium	Sprint-1
Software type Developer	Software used priors	USN-11	In User point, the Software Used must be in the updated version.	Version of the software used.(updated)	High	Sprint-1
Marketing type	Values of the application	USN-12	The rate of their values may Create an impressions by their values and reviews.	Share values of the application	Low	Sprint-2

CHAPTER -6

PROJECT PLANNING & SCHEDULING:

Project Planning & Design focuses on the schematic design phase of a project. In this experience area, you'll learn to lay out building designs, review building codes and regulations, coordinate schematics with consultants, and communicate design concepts with your client.

- Prepare Cost of Work estimates
- Determine impact of existing utilities infrastructure on site
- Apply principles of historic preservation for projects involving building restoration or renovation
- Understand implications of evolving sustainable design strategies and technologies
- Design landscape elements for site
- Define stakeholders. Stakeholders include anyone with an interest in the project. ...
- Define roles. Each stakeholder's role should be clearly defined. ...
- Introduce stakeholders.
- Set goals.
- Prioritize tasks.
- Create a schedule.
- Assess risks.
- Communicate.

6.1 SPRINT PLANNING & ESTIMATION:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	Creating Login page Creating Registration page	10	High	Karthick saran. Gopinath, Samson Durai
Sprint-1	Database Connectivity	USN-2	To Store details of the customer Connecting UI with Database	10	Medium	Ganapathy Sneha
Sprint-2	News Tracker UI	USN-3	Building UI News Tracker Application	10	High	Karthick saran Ganapathy Sneha
Sprint-2	API	USN-4	Connecting UI with News API, Google News API	10	High	Karthick saran Gopinath
Sprint-3	SendGrid Integration	USN-5	SendGrid Integration With Python Code	10	Low	Ganapathy Gopinath
Sprint-3	News Reader (Voice)	USN-6	Building Voice Assistant to read the news	10	Medium	Samson Durai, Sneha
Sprint-4	Containerization	USN-7	Containerizing the app	10	High	Ganapathy Karthick saran Sneha
Sprint -4	Upload image and deployment	USN-8	Upload Docker image to the IBM Registry and deploy it in the Kubernetes Cluster	10	High	Ganapathy , Gopinath Samson durai

Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

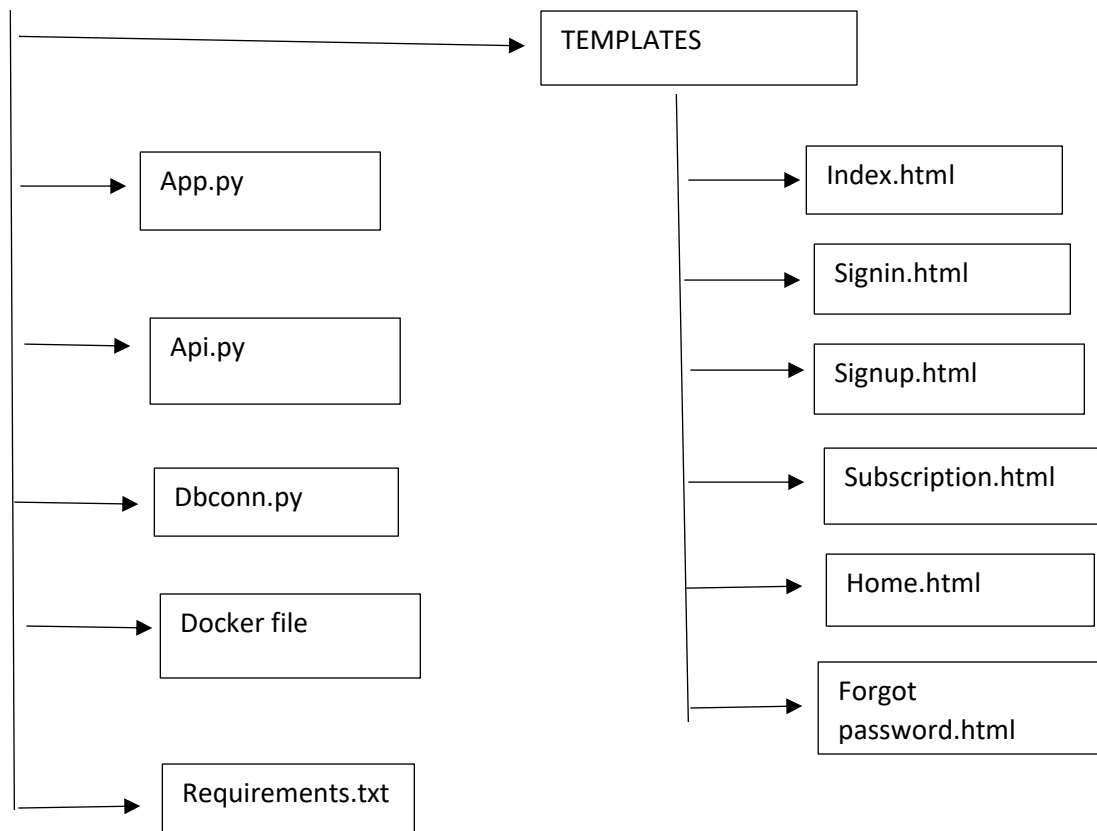
$$AV = \frac{\text{sprint duration}}{\text{velocity}} = \frac{20}{10} = 2$$

6.2 SPRINT DELIVERY SCHEDULE:

Title	Description	Date
Literature Survey and Information Gathering	Gathering Information by referring the technical papers, research publications etc	1 SEPTEMBER 2022
Prepare Empathy Map	To capture user pain and gainsPrepare List of Problem Statement	11 SEPTEMBER 2022
Ideation	Prioritize a top 3 ideas based on feasibility and Importance	18 SEPTEMBER 2022
Proposed Solution	Solution include novelty,feasibility,business model,social impact and scalability of solution	24 SEPTEMBER 2022
Problem Solution Fit	Solution fit document	1 October 2022
Solution Architecture	Solution Architecture	1 October 2022
Customer Journey	To Understand User Interactions and experienceswith application	9 October 2022
Functional Requirement	Prepare functional Requirement	15 October 2022
Data flow Diagrams	Data flow diagram	15 October 2022
Technology Architecture	Technology Architecture Diagram	16 October 2022
Project Development-Delivery of sprint 1,2,3 &4	Develop and submit the developed code by testing it	24 October 2022 – 19 November 2022

7.CODING & SOLUTIONING:

FOLDER STRUCTURE:



7.1 FEATURE -1:

Login


Email

Password

login

[Don't have an account](#) [Register](#)

[Forget password](#)



Register


Username

Email

Password

Sign Up

[You have an account](#) [login](#)



Login and Registration:

- In the beginning, Users can create the account in our app by filling the required information and necessary details.
- If already an account exist, the users can login through the account and gather information.

#registration page code

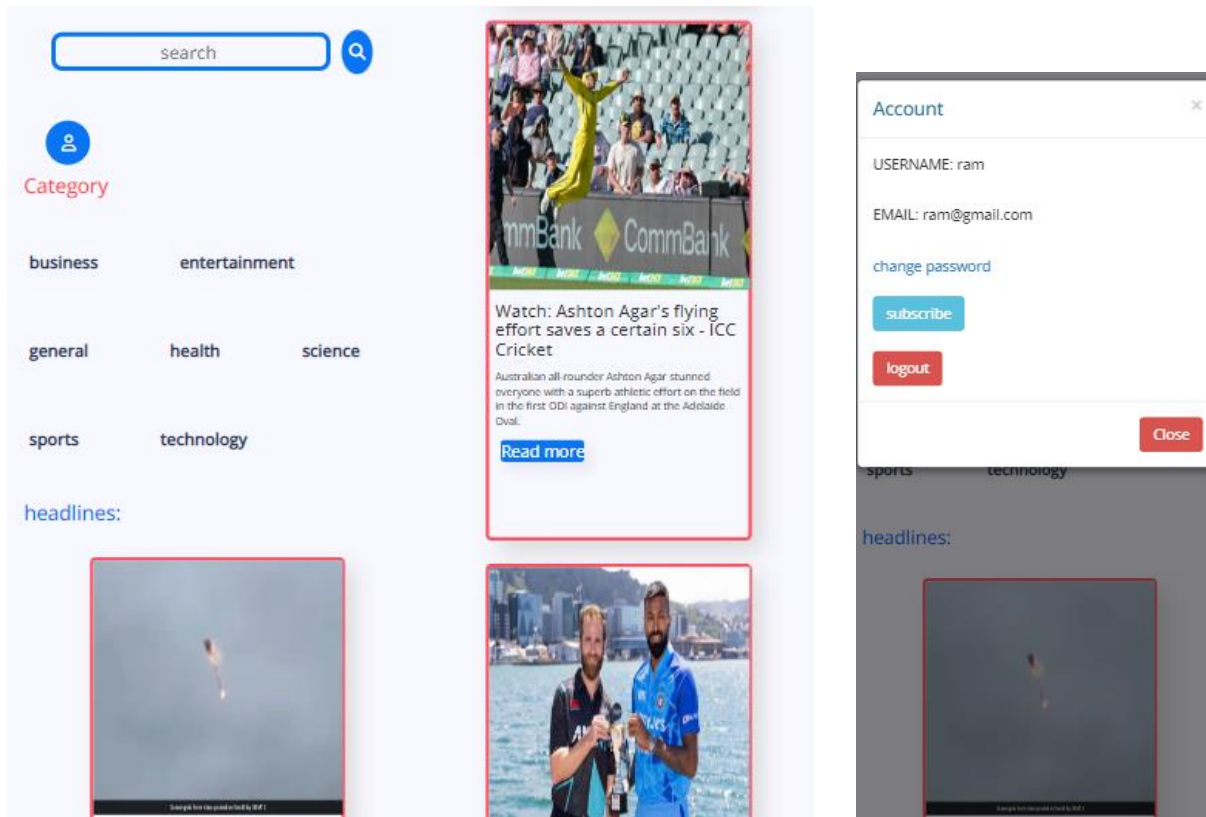
```
@app.route("/registration", methods=['POST'])
def signup():
    if request.method == 'POST':
        name = request.form.get('name')
        email = request.form.get('email')
        pwd = request.form.get('pwd')
        emailcheck="SELECT * FROM userdetails WHERE EMAIL='{0}' "
        smt = ibm_db.prepare(dbconn.con, emailcheck.format(email))
        ibm_db.execute(smt)
        mailres=ibm_db.fetch_assoc(smt)
        if mailres:
            return render_template("signup.html",msg="Email Is Already Taken")
        else:
            sql = "INSERT INTO userdetails (id,username,email,password,otp) VALUES ('{0}','{1}','{2}','{3}', '{4}')"
            res = ibm_db.exec_immediate(dbconn.con, sql.format(uuid.uuid4(),name, email, pwd,"123"))
            if sql:
                return redirect("/signin")
            else:
                return redirect("/")
        else:
            print("Could'nt store anything...")
```

#login page code

```
@app.route("/login", methods=['POST'])
def signin():
    if request.method == 'POST':
        email = request.form.get('email')
        pwd = request.form.get('pwd')
        sql = "SELECT * FROM userdetails WHERE EMAIL='{0}' AND PASSWORD='{1}'"
        smt = ibm_db.prepare(dbconn.con,sql.format(email,pwd))
        ibm_db.execute(smt)
        res=ibm_db.fetch_assoc(smt)

        if res:
            session["UID"]=res['ID']
            sql = "SELECT * FROM subscription WHERE EMAIL='{0}'"
            smt = ibm_db.prepare(dbconn.con, sql.format(email))
            ibm_db.execute(smt)
            res=ibm_db.fetch_assoc(smt)
            if res:
                session["sub"]=True
            else:
                session["sub"]=False
            return redirect("/home/api/headlines?page=1")
        else:
            return render_template("signin.html",msg="Invalid Email Or Password")
    else:
        print("Could'nt store anything...")
```

HOME PAGE:



HOME:

- In home page, Current Top-Headlines will appear and with various topics.
- Different category of news will appear such as health, sports, science, technology,
- Entertainment.
- In the right Top- corner user can access with their own account details such as username, Email and Logout option.

```

#home page

@app.route('/home/api/headlines')
def home():
    if "UID" in session:
        sql = "SELECT * FROM userdetails WHERE ID='{0}' "
        smt = ibm_db.prepare(dbconn.con, sql.format(session["UID"]))
        ibm_db.execute(smt)
        res=ibm_db.fetch_assoc(smt)
        if res:
            h=api.headlines()
            tore=h['totalResults']
            to=math.ceil(tore/int(9))
            u=request.base_url
            pn=request.args.get('page')
            pn=1
            if pn:
                e_p=int(pn)*int(9)
                s_p=int(e_p)-int(9)
            return render_template("home.html",data=res,news=h,url_head=u,s=s_p,e=e_p,
total=to,title="headlines")
        else:
            return redirect("/signin")
    else:
        return redirect("/signin")

```

```

#search module

@app.route('/home/api/query',methods=['GET'])
def query():
    if "UID" in session:
        sql = "SELECT * FROM userdetails WHERE ID='{0}' "
        smt = ibm_db.prepare(dbconn.con, sql.format(session["UID"]))
        ibm_db.execute(smt)
        res=ibm_db.fetch_assoc(smt)
        if res:
            q=request.args.get('search')
            s=api.search(q)
            tore=s['totalResults']
            to=math.floor(tore/int(9))
            if to>=10:
                to=10
            u=request.base_url
            pn=request.args.get('page')
            pn=1
            if pn:
                e_p=int(pn)*int(9)
                s_p=int(e_p)-int(8)
            return render_template("home.html",data=res,news=s,url_query=u,q=q,s=
s_p,e=e_p,total=to,title=q)
        else:
            return redirect("/signin")
    else:
        return redirect("/signin")

```

```

#category module

@app.route('/home/api/cat-list',methods=['GET'])
def category():
    if "UID" in session:
        sql = "SELECT * FROM userdetails WHERE ID='{0}' "
        smt = ibm_db.prepare(dbconn.con, sql.format(session["UID"]))
        ibm_db.execute(smt)
        res=ibm_db.fetch_assoc(smt)
        if res:
            q=request.args.get('category')
            s=api.category(q)
            tore=s['totalResults']
            to=math.floor(int(tore)/int(9))
            print(to)
            u=request.base_url
            pn=request.args.get('page')
            if pn:
                e_p=int(pn)*int(9)
                s_p=int(e_p)-int(9)
                return render_template("home.html",data=res,news=s,url_cat=u,s=s_p,e=
e_p,qu=q,total=to,title=q)
            else:
                return redirect("/signin")
        else:
            return redirect("/signin")
    else:
        return redirect("/signin")

```

7.2 FEATURE-2:

SUBSCRIPTION:

Get your Favourite category Updates
Delivered to You Daily

Receive the free newsletter millions of subscribers read every morning.

pincode:

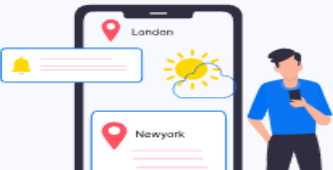
location:

Email:

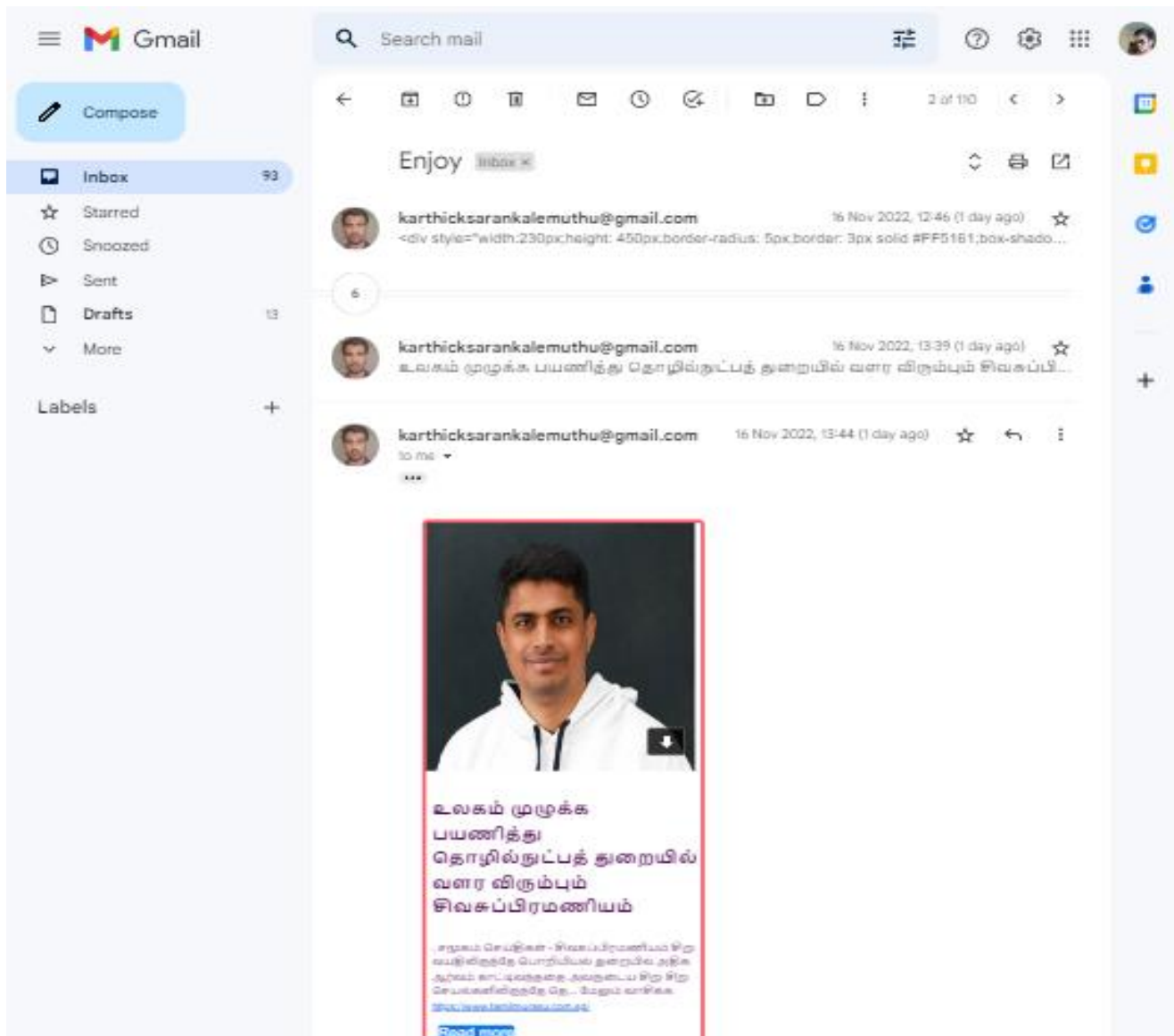
Category:

Languages:

subscribe



- The Subscription availability is based upon the users own criteria and wish list in accordance with their needs.
- This includes Pin-code, Location, Email, Category, Languages and fill these fields with necessary details and receive an instant mail notification.



```

#subscription module

@app.route("/sub-email", methods=['POST'])
def sub():
    if request.method == 'POST':
        loca= request.form.get('location')
        email = request.form.get('email')
        cat = request.form.get('cat')
        lang= request.form.get('lang')
        usercheck="SELECT * FROM userdetails WHERE ID='{0}' "
        smt = ibm_db.prepare(dbconn.con, usercheck.format(session["UID"]))
        ibm_db.execute(smt)
        mailres=ibm_db.fetch_assoc(smt)
        if mailres:
            sql =
"INSERT INTO Subscription (id,email,location,category,language) VALUES ('{0}','{1}','{2}'
','{3}','{4}')"
            res = ibm_db.exec_immediate(dbconn.con, sql.format(session["UID"],email,
loca,cat,lang))
            return redirect("/home/api/headlines?page=1")
        else:
            return print("Could'nt store anything...")

    else:
        return render_template("signup.html",msg="Register and enjoy subscription")

```

```

#send news on mail module

@app.route('/send')
def send():
    if "sub" in session:
        sql = "SELECT * FROM SUBSCRIPTION WHERE ID='{0}'"
        smt = ibm_db.prepare(dbconn.con, sql.format(session["UID"]))
        ibm_db.execute(smt)
        res=ibm_db.fetch_assoc(smt)
        r="https://gnews.io/api/v4/top-headlines?token={0}&topic={1}&lang={2}
&country=in&max=10"
        key="9f69b1d56f54a24b8fd1e51b392048de"
        s=r.format(key,res["CATEGORY"],res["LANGUAGE"])
        d=requests.get(s)
        data=d.json()
        sql1 = "SELECT * FROM userdetails WHERE ID='{0}'"
        smt1 = ibm_db.prepare(dbconn.con, sql1.format(session["UID"]))
        ibm_db.execute(smt1)
        res1=ibm_db.fetch_assoc(smt1)
        msg = Message( 'Read Everyday',sender ='karthicksarankalemuthu@gmail.com',recipients
=[res1["EMAIL"]])
        news=
'<div style="width:230px;height: 550px;border-radius: 5px;border: 3px solid #FF5161;box-shadow: 10px
x 10px 20px gainsboro;margin:20px;"><div><h4 style=" font-size: 15px;margin-left: 5px;overflow: hidden;">'+
'{1}'+
'</h4><p style=" font-size: 9px;font-weight: 500;margin-left: 5px;word-wrap: break-word;overflow: h
idden;">'+
+'{2}'+</p><a href="'+{3}'+
'" style=" width:140px;height: 35px;border-radius:3px;outline: none;border: none; background-colo
r:#0975F3;color: #fff;box-shadow: 6px 7px 10px #E7E9F6; text-decoration: none;margin: 10px;">Read m
ore</a><br></div></div>'
        sendnews=news.format(data["articles"][0]["image"],data["articles"][0]["title"],data[
"articles"][0]["description"],data["articles"][0]["url"])
        msg.html=sendnews
        mail.send(msg)
        return redirect("/home/api/headlines?page=1")
    else:
        return redirect("/signin")

```

Forgot password:

Change password

OTP

Enter the OTP


password

Enter the password

confirm password

Enter the Confirm-password

reset



- Click the forgot password and pop-up will appear user enter the already registered mail id.
- To reset the password the random OTP will send to the respective mail and then, enter the six-digit OTP received.
- After, the user can reset the password.

```

#forget password send otp

@app.route('/forget-password', methods=['POST'])
def forgetpassword():
    if request.method == 'POST':
        email = request.form.get('email')
        otp=randint(000000,999999)
        msg = Message( 'OTP',sender = 'karthicksarankalemuthu@gmail.com',
recipients = [email])
        msg.body =str(otp)
        mail.send(msg)
        sql = "UPDATE userdetails SET OTP='{0}' WHERE EMAIL='{1}'"
        l=email.lower()
        res = ibm_db.exec_immediate(dbconn.con, sql.format(str(otp),l))
        if sql:
            return render_template("forgetpassword.html",Message=
"OTP is send your email")
        else:
            return render_template("signin.html")
    else:
        return render_template("signin.html")

```

```

#verifying otp

@app.route('/verify', methods=['POST'])
def verify():
    if request.method == 'POST':
        otp = request.form.get('OTP')
        pwd = request.form.get('pwd')
        sql="SELECT * FROM userdetails WHERE OTP='{0}' "
        smt = ibm_db.prepare(dbconn.con, sql.format(otp))
        ibm_db.execute(smt)
        res=ibm_db.fetch_assoc(smt)
        if res:
            sql = "UPDATE userdetails SET PASSWORD='{0}' WHERE OTP='{1}'"
            res = ibm_db.exec_immediate(dbconn.con, sql.format(pwd,otp))
            return render_template("signin.html",Message=
"password changed successfully")
        else:
            return render_template("forgetpassword.html",Message=
"OTP is invalid")

```

DBCONNECTION:

```
import ibm_db
try:
    con = ibm_db.connect(
        "DATABASE=bludb;HOSTNAME=b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=31249;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=hrc47660;PWD=vn0D4tZ0VkJyM6xw;"
        , '', '')
    print("db connection successfully")
except:
    print("db connection failed")
```

FETCH API:

```
import requests
import os

key=os.environ.get('news_api')

def headlines():
    r="https://newsapi.org/v2/top-headlines?country=in&apiKey={}"
    s=r.format(key)
    d=requests.get(s)
    data=d.json()
    return data

def search(q):
    r="https://newsapi.org/v2/everything?q={}&apiKey={}"
    s=r.format(q,key)
    d=requests.get(s)
    data=d.json()
    return data

def category(cat):
    r="https://newsapi.org/v2/top-headlines?country=in&pageSize=100&category={}&apiKey={}"
    s=r.format(cat,key)
    d=requests.get(s)
    data=d.json()
    return data
```

CSS:

```
@import  
url('https://fonts.googleapis.com/css2?family=Open+Sans:wght@50  
0&display=swap');  
*{  
    padding: 0px;  
    margin: 0px;  
    box-sizing: border-box;  
}  
body{  
    background-color: ghostwhite !important;  
    font-family: 'Open Sans', sans-serif !important;  
}  
.navbar{  
    background-color: transparent !important;  
}  
.nav li a{  
    margin:20px !important;  
    color:#293559 !important;  
    font-size: 15px !important;  
    font-weight:600 !important;  
}
```

```
.nav li a:hover{  
    background-color: transparent !important;  
}  
#text h1{  
    color:#0975F3 !important;  
    font-size: 25px !important;  
    font-weight:500 !important;  
}  
#text h4{  
    color:#293559;  
}  
#text{  
    margin-top: 60px !important;  
    margin-left: 15px;  
}  
#text h1,#text h4{  
    margin: 27px;  
}  
#signup,#start{  
    color:#ffffff !important;  
    background-color:#0975F3!important;  
    border-radius:10px !important;  
    margin: 25px !important;
```

```
padding: 7px 35px !important;
font-size: 15px !important;
font-weight:900 !important;
}

.navbar-brand{
margin:15px !important;
color:#0975F3 !important;
font-size: 25px !important;
font-weight:600 !important;
}

span{
background-color:#0975F3 !important;
width: 40px !important;
height: 6px !important;
}

@media screen and (max-width:600px){
#text h1{
font-size: 15px !important;
font-weight:300 !important;
}
#text{
margin-top: 20px !important;
margin-left: 10px;
```



```
}
```

```
#text h4{
```

```
    font-size: 10px !important;
```

```
}
```

```
#signup,#start{
```

```
    color:#ffffff !important;
```

```
    background-color:#0975F3!important;
```

```
    border-radius:5px !important;
```

```
    margin: 25px !important;
```

```
    padding: 5px 17px !important;
```

```
    font-size: 10px !important;
```

```
    font-weight:400 !important;
```

```
}
```

```
.navbar-brand{
```

```
    margin: 5px !important;
```

```
    color:#0975F3 !important;
```

```
    font-size: 15px !important;
```

```
    font-weight:400 !important;
```

```
}
```

```
span{
```

```
    background-color:#0975F3 !important;
```

```
    width: 20px !important;
```

```
    height: 3px !important;
```

```
}  
#img{  
    width: 300px !important;  
    height: 300px !important;  
}  
}
```

```
form h3{  
    margin-left: 150px;  
    color:#0975F3;  
}  
form{  
    display: flex;  
    flex-direction:column;  
}  
input,select{  
    width:400px !important;  
    height: 45px;  
    border-radius: 10px;  
    outline: none;  
    border: none;
```

```
background-color: #E7E9F6;  
text-align: center;  
font-size: 35px;  
font-size: 600;  
}  
input::placeholder{  
    text-align: center;  
    color:gray;  
}  
form label{  
    color: #8F8AA8;  
    font-size: 25px;  
    font-weight: 800;  
}  
#submit{  
    width:250px;  
    height: 45px;  
    border-radius: 10px;  
    outline: none;  
    border: none;  
    background-color:#0975F3;  
    color: #fff;  
    border-radius: 10px;
```

```
    font-size: 20px;
    box-shadow: 3px 3px 5px #E7E9F6;
}
```

```
#sign{
    display: inline-flex;
}
```

```
#signin{
    margin-left: 70px;
    margin-top: 10px;
    color: #0975F3;
    font-size: 20px;
    font-weight: 400;
    text-decoration: none;

}
```

```
#detiles{
    display: flex;
    flex-direction: row;
    margin: 15px;
}
```

```
#detiles h5{
    color: #171717;
    font-size: 20px;
```

```
    font-weight: 40px;
    margin-right: 30px;
}
#detiles .a{
    font-size: 20px;
    font-weight: 40px;
    text-decoration: none;
    margin:6px;
}
#signup_container{
    margin-top: 30px !important;
    margin-left: 20px;
}
#sub-container{
    margin-top: 30px !important;
    margin-left: 40px;
}
#signin_container{
    margin-top: 70px !important;
    margin-left: 40px;
}
#gear1,#gear2,#gear1_2,#gear1_3,#gear2_2{
    animation: move 1s infinite;
```

```
transform-box: fill-box;  
transform-origin:center;  
}
```

```
@keyframes move {  
  10%{  
    transform: rotate(90deg);  
  }  
  50%{  
    transform: rotate(180deg);  
  
  }  
  100%{  
    transform: rotate(360deg);  
  }  
}
```

```
@media screen and (max-width:600px){  
  input{  
    width:150px;  
    height: 35px;  
  }  
  form{  
    margin-left: 0px;
```

}

form h3{

margin-left:20px;

}

#sign #submit{

width:150px;

height: 35px;

}

#signin{

margin-left:35px;

margin-top: 10px;

}

#detiles h5{

font-size: 15px;

font-weight:300;

margin-right: 15px;

}

#detiles .a{

font-size: 15px;

font-weight:300px;

text-decoration: none;

margin:7px;

}

}

```
#search_container{  
    display: flex;  
    flex-direction: row;  
    margin: 15px;  
    align-items: center;  
    justify-content: center;  
}
```

```
#searchbar{  
    width:550px !important;  
    height: 35px;  
    border-radius: 20px;  
    outline: none;  
    border: 4px solid #0975F3;  
    background-color: transparent;  
    font-size: 16px;  
    font-size: 400;  
    margin: 10px;  
}
```



```
/*#home_container{  
    display: flex !important;  
    flex-direction: row !important;  
    margin: 15px !important;  
    justify-content:space-evenly !important;  
}*/
```

```
#user{  
    width: 40px;  
    height: 40px;  
    border-radius: 50%;  
    background-color: #0975F3;  
    color: white;  
    font-weight: 800;  
    border: none;  
    margin-top: 20px !important;  
    margin-left: 450px !important;  
    outline: none;  
}
```

```
#searchbtn{  
    width: 40px;  
    height: 40px;  
    border-radius: 50%;
```

```
background-color: #0975F3;  
color: white;  
font-weight: 800;  
border: none;  
outline: none;  
}  
#card{  
    width: 230px;  
    height: 450px;  
    border-radius: 5px;  
    border: 3px solid #FF5161;  
    box-shadow: 10px 10px 20px gainsboro;  
    margin:20px;  
}  
#card #card-img{  
    width: 225px;  
    height: 230px;  
}  
#card #card-title{  
    font-size: 15px;  
    margin-left: 5px;  
    overflow: hidden;  
}
```

```
#card #card-text{  
    font-size: 9px;  
    font-weight: 500;  
    margin-left: 5px;  
    word-wrap: break-word;  
    overflow: hidden;  
}
```

```
#readbtn{  
    width:140px;  
    height: 35px;  
    border-radius:3px;  
    outline: none;  
    border: none;  
    background-color:#0975F3;  
    color: #fff;  
    box-shadow: 6px 7px 10px #E7E9F6;  
    text-decoration: none;  
    margin: 10px;  
}
```

```
#cate li{  
    display: inline-block;  
}
```

```
#cate li a{
```

```
    text-decoration: none;
}

.nav .navbar-nav li .active{
    background-color: transparent !important;
    color: #FF5161 !important;
}

#pagination{
    margin-left: 230px !important;
}

@media screen and (max-width:600px){
    #searchbar{
width:350px !important;
height: 35px;
border-radius: 10px;
font-size: 16px;
margin: 10px;
    }
    #user{
        width: 40px;
        height: 40px;
        border-radius: 50%;
        background-color: #0975F3;
        color: white;
    }
}
```

```
    font-weight: 500;
    border: none;
    margin-left: 20px !important;
    outline: none;
}

#card{
    width: 230px;
    height: 450px;
    border-radius: 5px;
    border: 3px solid #FF5161;
    box-shadow: 10px 10px 20px gainsboro;
    margin: 20px;
    margin-left: 60px;
}

#pagination{
    margin-left: 80px !important;
}

}

.likebtn{
    fill: #FF5161;
}

#social{
```

```
    display: inline-block;  
}
```

```
#social i{  
    margin: 10px;  
    width: 35px;  
    height: 35px;  
}
```

```
#radio{  
    width: 20px;  
    height: 20px;  
}
```

7.3 DATABASE SCHEMA:

TABLE:

- User details
- Subscription.

USER DETAILS:

NAME	DATATYPE	NULLABLE	LENGTH
S NO	INTEGER	N	
ID	VARCHAR	N	255
USER NAME	VARCHAR	N	255
EMAIL	VARCHAR	N	255
PASSWORD	VARCHAR	N	255
OTP	VARCHAR	N	255

```
CREATE TABLE USERDETAILS(  
    sno INT GENERATED ALWAYS AS IDENTITY NOT NULL,  
    id VARCHAR(255) NOT NULL,  
    username VARCHAR(255) NOT NULL,  
    email VARCHAR(255) NOT NULL,  
    password VARCHAR(255) NOT NULL,  
    otp VARCHAR(255) NOT NULL,  
    PRIMARY KEY (sno,id)  
);
```

SUBSCRIPTION:

NAME	DATATYPE	NULLABLE	LENGTH
S NO	INTEGER	N	
ID	VARCHAR	N	255
EMAIL	VARCHAR	N	255
LOCATION	VARCHAR	N	255
CATEGORY	VARCHAR	N	255
LANGUAGE	VARCHAR	N	255

```
CREATE TABLE SUBSCRIPTION(  
    sno INT GENERATED ALWAYS AS IDENTITY NOT NULL,  
    id VARCHAR(255) NOT NULL,  
    email VARCHAR(255) NOT NULL,  
    location VARCHAR(255) NOT NULL,  
    category VARCHAR(255) NOT NULL,  
    language VARCHAR(255) NOT NULL,  
    PRIMARY KEY (sno,id)  
);
```


8.TESTING:

8.1 TEST CASES:

Test Scenarios

1	verify user is able to see the Login page when user clicked on login button
2	Verify the UI elements in Login/Signup page
3	Verify user is able to log into application with Valid credentials
4	Verify user is able to log into application with InValid credentials
5	verify user is able to sign up the valid email account
6	verify user is able to sign up the Invalid email account
7	verify the headlines news will be displayed or not?
8	verify the selected category news displayed or not?
9	verify the search results are displayed or not?
10	Verifying the user details displayed or not?
11	Verify the Random OTP send or not?
12	verify the valid OTP and password and confirm password must be same
13	verifying the user can receive mail or not in our favourite category
14	verify user can logged out or not

8.2 USER ACCEPTANCE TESTING:

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status
LoginPage_TC_001	Functional	Home Page	verify user is able to see the Login page when user clicked on login button		1.Enter URL and click go 2.Click on the login button 3.Verify login page displayed or not	http://169.51.200.171:3032/4/	Login page should displayed	Working as expected	Pass
LoginPage_TC_002	UI	Home Page	Verify the UI elements in Login/Signup page		1.Enter URL and click go 2.Click on login button 3.Verify login page with below UI elements: a.email text box b.password text box c.Login button d. Create new account link e.Forgot password? Recovery password link	http://169.51.200.171:3032/4/signup	Application should show below UI elements: a. email text box b. password text box c. Login button with blue colour d. Create new account link e. Forgot password? Recovery password link	Working as expected	pass
LoginPage_TC_003	Functional	Home page	Verify user is able to log into application with Valid credentials		1.Enter URL(http://169.51.200.171:3032/4/signup) and click go 2.Click on login in button 3.Enter Valid username/email in Email text box 4. Enter valid password in password text box 5.Click on login button	Username: ganapathys027@gmail.com password: 123	User should navigate to user account homepage	working as expected	pass
					1.Enter	Username:	Application should show		

9. RESULTS:

Users are usually overwhelmed by the regular news updates, especially if received during busy hours. Regular updates at unequal intervals of time by multiple apps tend to disturb and also spam the user's notifications bar thus causing him to feel annoyed and clearing which makes him miss out on relevant updates from other apps too.

This solves this problem by allowing the user to decide when he can receive articles. This way he will at least be more inclined to read the article and the irritating from spamming will be relieved.

9.1 Performance Metrics:

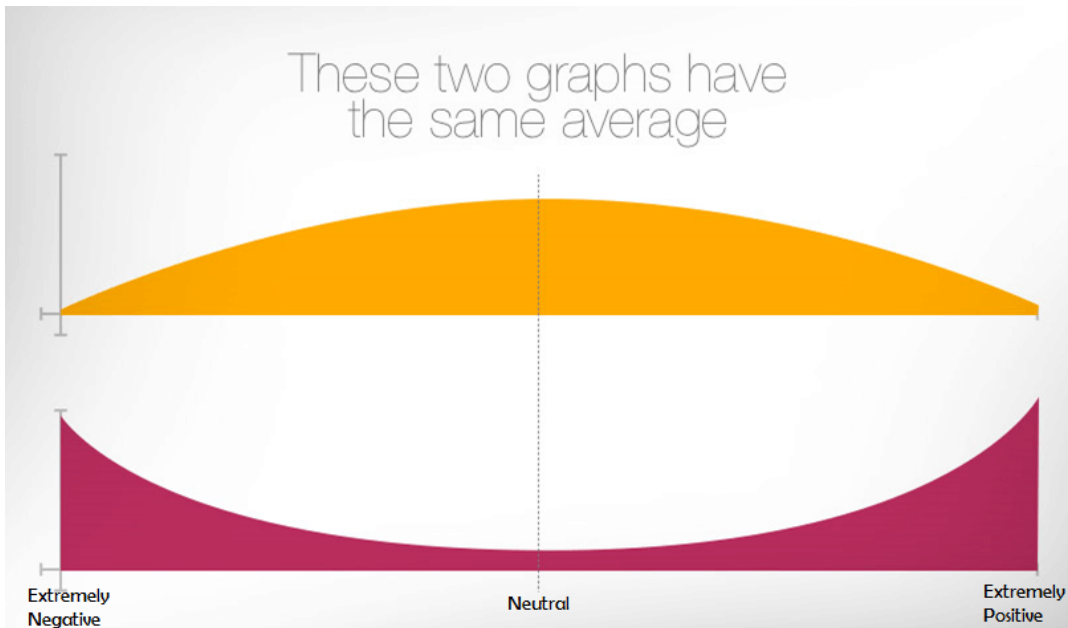
1. User Satisfaction Scores

The application performance index, score, has become an industry standard for tracking the relative performance of an application.

It works by specifying a goal for how long a specific web request or transaction should take.

2. Average Response Time

Let me start by saying that **averages suck**. I highly recommend using the aforementioned user satisfaction scores as a preferred way to track overall performance. That said, averages are still a useful application performance metric.



3. Error Rates

The last thing you want your users to see are errors. Monitoring error rates is a critical application performance metric.

There are potentially 3 different ways to track application errors:

- HTTP Error % – Number of web requests that ended in an error
- Logged Exceptions – Number of unhandled and logged errors from your application

4. Count of Application Instances

If your application scales up and down in the cloud, it is important to know how many server/application instances you have running. Auto-scaling can help ensure your application scales to meet demand and saves you money during off-peak times. This also creates some unique monitoring challenges.

5. Request Rate

Understanding how much traffic your application receives will impact the success of your application. Potentially all other application performance metrics are affected by increases or decreases in traffic.

6. Application & Server CPU

If the CPU usage on your server is extremely high, you can guarantee you will have application performance problems. Monitoring the CPU usage of your server and applications is a basic and critical metric.

10.ADVANTAGES AND DISADVANTAGES:

ADVANTAGES:

There are many sites that provide free app for iPhone, Android, and other platforms. The user gets free app downloads from the websites. There are many features in the applications that give live details to the user.

Home screen widgets: The user can set the home screen widgets of these apps that help in quick access of the application and stories. From there the user can easily read the news in a single click.

These applications also allow to post comments to the users on the stories.

Automatic updates: These applications provide the latest information about the world. The user doesn't need to refresh the app window, they get automatically updated with the time. The user can get regular updates of sports, lifestyle, business, and other things on their device with easy navigation.

Live articles & coverages: These apps provide the latest articles and coverages of news on the device. The user can browse or watch live videos through their devices.

These applications give access to all the current or live details in a single tap on the home screen.

Offline reads(mail): The user can read the news or view the results offline with the apps. They store the favorite topics and bookmarks in the device memory for future use. More so, the user can post comments on the news articles after logging in the application.

Push notifications(mail): These apps provide push notifications when the user is offline or opening any other window on the device. These notifications display in the notification panel of the device and the user can directly access the stories from there.

DISADVANTAGES:

Disadvantages. Development time and cost is much higher than optimizing a website and creating a WebApp. To make application available to the whole market you will also need to develop Apps for the Android and Blackberry platforms. Code cannot be reused across the platforms.

1. Mobile apps don't substitute a website.
2. You'll need Android and iOS applications and listings.
3. Update and maintenance efforts are multiplied.
4. There is extra Marketing pressure too.
5. You need a separate URL.
6. Native applications don't help with SEO.

Is it better to use apps or websites?





Analysis shows that the applications are more popular than equivalent websites, as they're more convenient. Mobile apps provide better user experiences, load content faster, and are easier to use. Besides, unlike websites, apps have push notifications.

11.CONCLUSION:

From the results of the validation, it can be concluded the design that I proposed can further enhance user experience in using Trading news, but it still needs to be iterated for several points.

Feedback from Participant

- Mostly all participants satisfied with the new design
- They said the top bar is really helpful
- The new design more organized and provide more information.
- The user experience for ads free usage.

Point	Before	After
User easily to navigate and do interaction		
User easiliy get information and do action in detail news		

12. FUTURE SCOPE:

1. Progressive Web Apps (PWAs)

Progressive Web Apps (PWAs) are cross-platform web applications that take advantage of advances in web technology to provide a native app-like user experience. In comparison to complex mobile apps, PWAs are simple to create.

PWAs are a hybrid of conventional web pages and native apps in their most basic form. Standard web technologies like HTML, CSS, JavaScript, Angular, and React can be used to create Progressive Web Apps. These web apps are simple to create and offer an excellent user experience on both desktop and mobile devices.

2. Motion UI

It's difficult to attract users to a website that solely contains information and HTML code. The front-end of today's websites must be more entertaining, intuitive, and user-friendly. As a result, websites or applications can capture the attention of customers and pique their curiosity.

3. Blockchain Technology

Blockchain is a decentralized system that enables the transmission of digital assets via the internet. Blockchain isn't the most recent web development innovation. Stuart Haber and W Scott Stornetta first described it in 1991 as a cryptographically secured chain of blocks.

Bitcoin, based on blockchain technology, was founded in 2009 by an unknown person under the pseudonym Satoshi Nakamoto, and quickly gained popularity as technology progressed. As a result, blockchain has emerged as the web's future technology, and various businesses have begun to construct a decentralized system.

4. Cybersecurity

As people's reliance on the internet grows, most of their daily routine operations rely on it, whether it's banking, paying bills online, sending data, talking, or shopping. There's always the risk of data being stolen.

As a result, the release of advanced cybersecurity technology in web applications helps businesses and organizations to design incredibly secure websites, apps, or servers that can thwart cyber-attacks such as data breaches, phishing, denial-of-service (DoS), SQL Injection, or Malware.

5. API-first Approach

The requirement for APIs based on the API-first strategy that can run efficiently on all devices, platforms, and operating systems is growing rapidly for Web Applications. It prefers to start with APIs and then switch to a code-first approach.

Microservice Architecture-based API-first approach enables web designers to create application programming interfaces (APIs) before they are implemented. Microservices-based application interfaces are highly manageable, tested, loosely connected, and dependably deployable. They give a pleasant user experience (UX) on all devices.

6. Single Page Applications

Users nowadays value their time. It might be aggravating for a user if a website takes a long time to load. As a result, single-page websites are presently in high demand and are the wave of the future in website building. It's also useful for companies that don't require a lot of material on their website.

It has gained popularity over MPAs after the introduction of SPA frameworks and AJAX (Multi-Page Applications). To load a large number of pages, MPAs must send numerous requests from the client to the server, which takes a long time (compared to SPAs). As a result, the developers were on the lookout for a different solution. It has become the first choice since the debut of JavaScript frameworks.

13.APPENDIX:

<https://github.com/IBM-EPBL/IBM-Project-15088-1659594117/>

The screenshot shows the GitHub interface for the repository `IBM-EPBL / IBM-Project-15088-1659594117`. The repository is public and has 3 stars, 1 watch, and 0 forks. The main content area displays the repository structure with a table of files and folders. The table has three columns: file/folder name, relative path, and last commit time. The files listed are `Assignments` (assignment, 5 days ago), `Project Design and planning` (sprint planning, 27 days ago), and `project development phase` (sprint-4, 4 hours ago). The repository is currently on the `master` branch with 1 branch and 0 tags. The commit hash is `e2b7f3b` and it was committed 4 hours ago. There are 38 commits in total. A green button labeled `<> Code` is visible. Below the table, there is a light blue box with the text "Help people interested in this repository understand your project by adding a README." and a green button labeled `Add a README`. On the right side, there are sections for `About` (News Tracker Application, 3 stars, 1 watching, 0 forks), `Releases` (No releases published, [Create a new release](#)), and `Packages` (No packages published, [Publish your first package](#)).

File/Folder	Path	Last Commit
Assignments	assignment	5 days ago
Project Design and planning	sprint planning	27 days ago
project development phase	sprint-4	4 hours ago

