NAALAIYA THIRAN PROJECT - 2022
19ECI01-PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

INVENTORY MANAGEMENT SYSTEM FOR RETAILERS

A PROJECT REPORT

Submitted by

TEAM ID: PNT2022TMID52669

ARUN PRAVIN	**1905069**
KARTHEESWARAN	**1905089**
THIRUMALAI SELVAN	**1905120**
JOEL J	**2005208**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COIMBATORE INSTITUTE OF TECHNOLOGY, COIMBATORE – 641 014

(Government Aided Autonomous Institution affiliated to Anna University)

ANNA UNIVERSITY: CHENNAI 600025

NOVEMBER 2022

PROJECT CALENDAR

Phase	Phase Description	Week	Dates	Activity Details
1	Preparation Phase (Pre- requisites, Registrations, Environment Set-up, etc.)	2	22 - 27 Aug 2022	Creation GitHub account & collaborate with Project repository in project workspace
2	Ideation Phase (Literature Survey, Empathize, Defining Problem Statement, Ideation)	2	29 Aug – 3rd Sept 2022	Literature survey (Aim, objective, problem statement and need for the project)
		3	5 - 10th Sept 2022	Preparing Empathy Map Canvas to capture the user Pains & Gains
		4	12 - 17 Sept 2022	Listing of the ideas using brainstorming session
3	Project Design Phase -I (Proposed Solution, Problem- Solution Fit, Solution Architecture)	5	19 - 24 Sept 2022	Preparing the proposed solution document
		6	26 Sept - 01 Oct 2022	Preparing problem - solution fit document & Solution Architecture
4	Project Design Phase -II (Requirement Analysis, Customer Journey, Data Flow Diagrams, Technology Architecture)	7	3 - 8 Oct 2022	Preparing the customer journey maps
		8	10 - 15 Oct 2022	Preparing the Functional Requirement Document & Data- Flow Diagrams and Technology Architecture
5	Project Planning Phase (Milestones & Tasks, Sprint Schedules)	9	17 - 22 Oct 2022	Preparing Milestone & Activity List, Sprint Delivery Plan
6	Project Development Phase (Coding & Solutioning, acceptance Testing, Performance Testing)	10	24 - 29 Oct 2022	Preparing Project Development - Delivery of Sprint-1
		11	31 Oct - 5 Nov 2022	Preparing Project Development - Delivery of Sprint-2
		12	7 - 12 Nov 2022	Preparing Project Development - Delivery of Sprint-3
		13	14 - 19 Nov 2022	Preparing Project Development - Delivery of Sprint-4

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 PROJECT OVERVIEW

The main objective of this project is to provide people of all age groups right from a high-schooler, entrepreneur to a businessman, an easy-to-use web application via which those people can access all types of news ranging from entertainment to business and politics in a one single place. Throught this web application, anyone, from anywhere in the world, at anytime can gain knowledge of the day-to-day happening of the world and their own surrounding in one single touch of the screen. Moreover, this application is like one click to gain knowledge in and around the world.

1.2 PURPOSE

The main purpose developing this web application is to eliminate any barrier in accessing or getting to know the day-to-day happenings. This application will provide all the day-to-days updates about various categories such as Entertainment, Sports, Business, Politics etc.., in one single place. Hence, saves time and the effort of the user to search for different categories separately, also making this web application an efficient eventually. Also, main purpose of this application terminates any possible information redundancy that may cause while surfing the news.

CHAPTER 2
LITERATURE SURVEY

2.1 EXISTING PROBLEM

Most of the people get the information about the world news through the internet, which is fast accessible and reliable. People have no time to be updated through newspaper or watching news in the television, so different web applications have introduced to provide news across the world. Most of the existing systems provides news only on some of the categories. Few single websites only provide local or national news. Most of the system does not integrate local, national and international news embedded in one system.

REFERENCE

1. Exploring mobile news reading interactions for news app personalisation – Marios Constantinde & John Dowell.
2. New technology, old practices: Examining news websites from a professional perspective - Itai Himelboim & Steve McCreery
3. The future of personalization at news websites: Lessons from a longitudinal study - Neil Thurman & Steve Schifferes
4. Detection and Tracking in News Articles – Sagar Patel, Sanket suthar.

2.2 PROBLEM STATEMENT DEFINITION

Most of the people do not like to carry a newspaper with them. Some people want them to be updated only in the area they are interested in. Users are not possible to get latest news in their busy schedule and a lot of fake news is broadcast. A news sharing app seeks to make it easier for the users to relevant news and to clearly indicate that the information is real and has been gathered from the reputable source.

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

An empathy map is a collaborative visualization used to express clearly what one knows about a particular type of user. It externalizes knowledge about users in order to createa shared understanding of user needs, and aid in decision making.

Empathy maps are split into 4 quadrants (Says, Thinks, Does, and Feels), with the userin the middle. Empathy maps provide a glance into who a user is as a whole. The ***Says*** quadrant contains what the user says or what he needs. The ***Thinks*** quadrant captures what the user is thinking throughout the experience. The ***Does*** quadrant encloses the actions the user takes. The ***Feels*** quadrant is the user's emotional state.

The empathy map for Inventory management system for retailers is shown in Fig 3.1

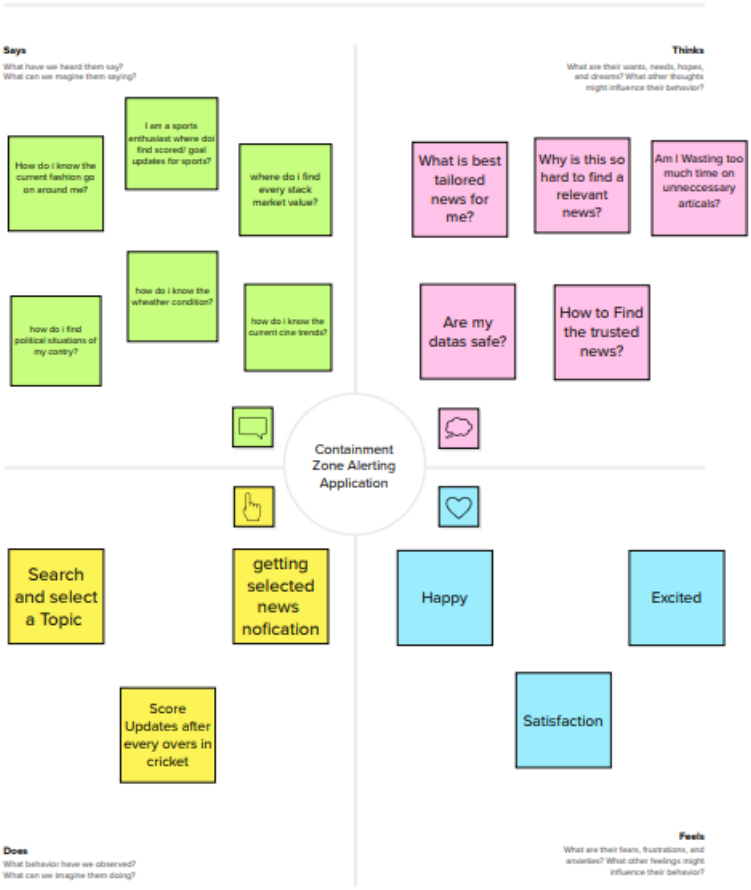


Fig 3.1 Empathy map

3.2 IDEATION AND BRAINSTORMING

Ideation is often closely related to the practice of brainstorming, a specific technique that is utilized to generate new ideas. Brainstorming is usually conducted by getting a group of people together to come up with either general new ideas or ideas for solving a specific problem or dealing with a specific situation. A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity. Both brainstorming and ideation are processes invented to create new valuable ideas, perspectives, concepts and insights, and both are methods for envisioning new frameworks and systemic problem solving.

The Ideation chart for Inventory management system for retailers is shown in Table 3.2.

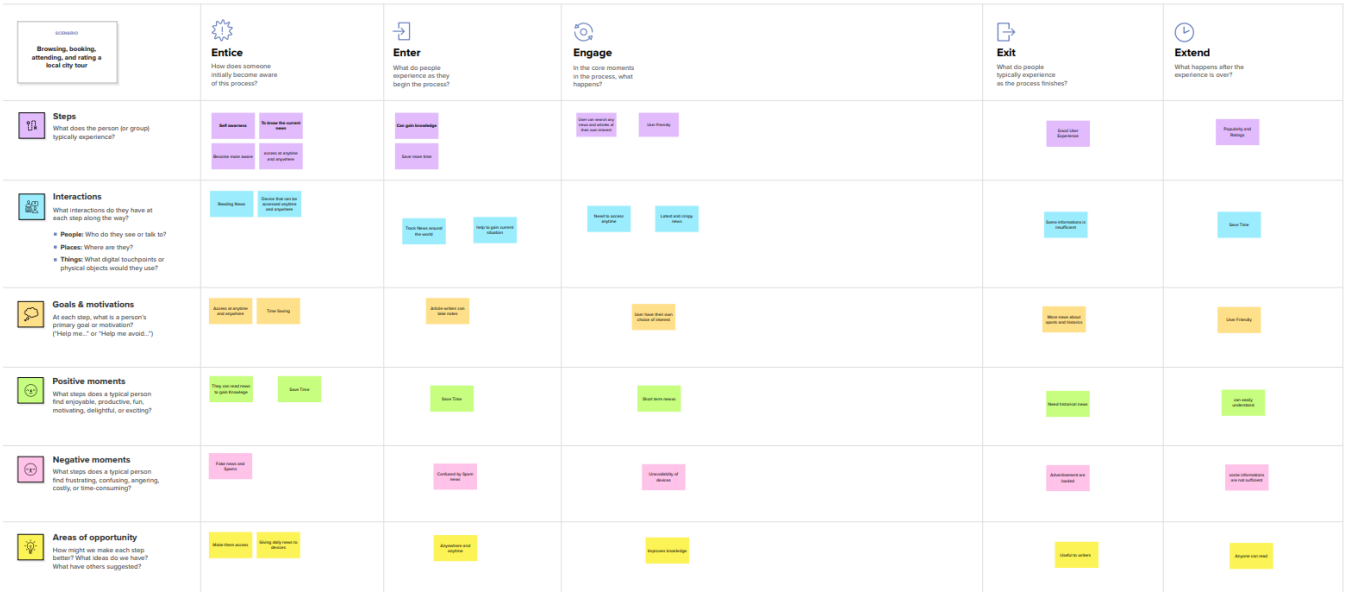


Table 3.2 Ideation and Brainstorming

3.3 PROPOSED SOLUTION

We are well aware of the user's interested and uninterested topics. By providing the news feed feature which contain quick short news where user time can be saved. An application needs to be developed in which users can read news whenever they want and they will be able to customize their area of interest. which aggregates the news from all the reliable sources and categorize them under different sections for the best user experience. As it was an application-based project, correct ideation and execution can develop an application with no bugs and errors, so that the user might like our application and some might suggest and share it to their surroundings, resulting in an increase in our application insights. This solution is scalable since it is enabled through cloud technologies offered by IBM enabling scalability on a consumer demand.

3.4 PROBLEM SOLUTION FIT

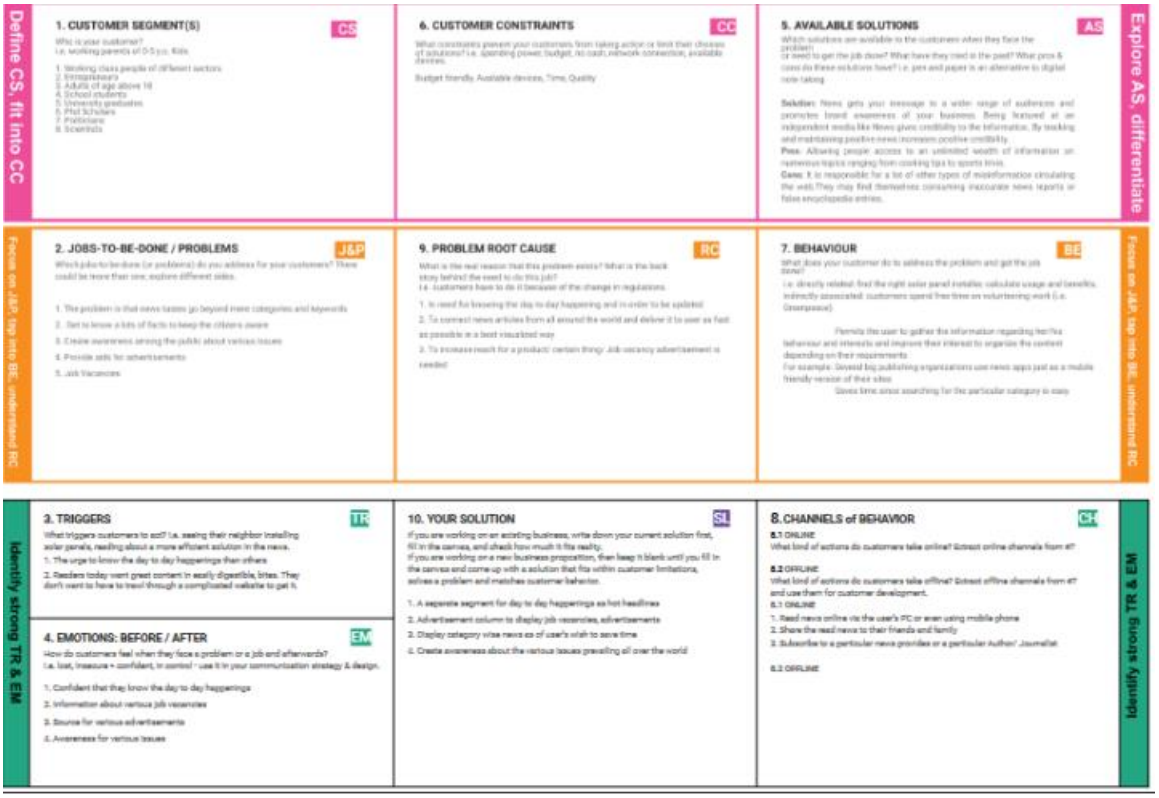


Fig 3.4 Problem Solution fit

CHAPTER 4

REQUIREMENT ANALYSIS

Requirements analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional and Non-functional requirements.

4.1 FUNCTIONAL REQUIREMENTS

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements. Following are the functional requirements of the proposed solution shown in table 4.1.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	User Installation	User can install the app from Google Play Store or from the website
FR-4	User Login	User should login the app with the user's name and password

Table 4.1 Functional Requirements

4.2 NON-FUNCTIONAL REQUIREMENTS

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements. Following are the non-functional requirements of the proposed solution shown in table 4.2.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Everyone can understand the process of using the app easily by the commands given in the app.
NFR-2	**Security**	It is a more secured app. No fake news can be shared.
NFR-3	**Reliability**	Easy to use at anytime and anywhere
NFR-4	**Performance**	Performance of the app is very great
NFR-5	**Availability**	More sub categories are available
NFR-6	**Scalability**	Efficiently manages many requests at once, delivering seamless experience to users.

Table 4.2 Non-Functional Requirements

CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

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 allthese years. While they work well for data flow software and systems, they are less applicablenowadays to visualizing interactive, real-time or database-oriented software or systems.
There are four main elements of a DFD — external entity, process, data store, and data flow.

External entity

An external entity, which are also known as terminators, sources, sinks, or actors, are an outside system or process that sends or receives data to and from the diagrammed system. They're either the sources or destinations of information, so they're usually placed on the diagram's edges. External entity symbols are similar across models except for Unified, which uses a stick-figure drawing instead of a rectangle, circle, or square.

Process

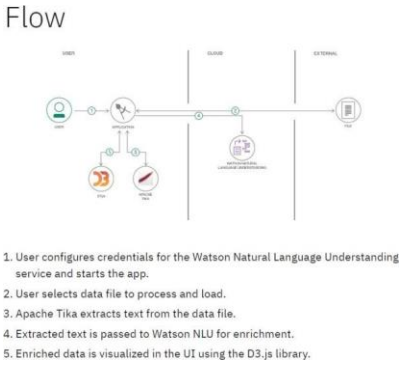
Process is a procedure that manipulates the data and its flow by taking incoming data, changing it, and producing an output with it. A process can do this by performing computations and using logic to sort the data, or change its flow of direction. Processes usually start from the top left of the DFD and finish on the bottom right of the diagram.

Data store

Data stores hold information for later use, like a file of documents that's waiting to be processed. Data inputs flow through a process and then through a data store while data outputs flow out of a data store and then through a process.

Data flow

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.



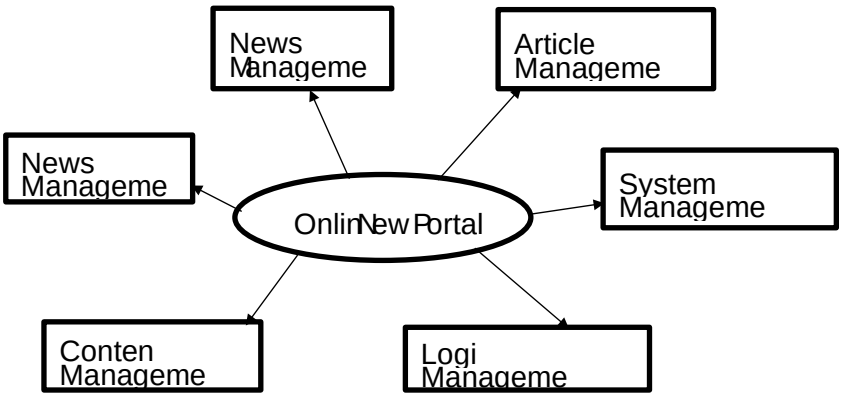


Fig 5.1: Data Flow Diagram

5.2 SOLUTION AND TECHNICAL ARCHITECHTURE

Solution architecture is a complex process with many sub-processes – that bridgesthe gap between business problems and technology solutions. A web application is designed to address the mentioned issues with the following functionalities: The web application will ask retailers to create their accounts by providing essential details. Retailers can access their accounts by logging into the application. Retailers can add products since it consumes less amount of time than manual entry. Once retailers successfully log in to the application they can update their inventory details, also users will be able to add new stock by submitting essential details related to the stock. They can view details of the current inventory. An alert is sent automatically by the inventory management system if the stock left count reaches a threshold value and as soon as the alert is received, the stocks required are ordered and as a result pausing of sale is avoided. A simple E-commerce web page is developed to check the stock management. We can order the new products from a particular retailer.

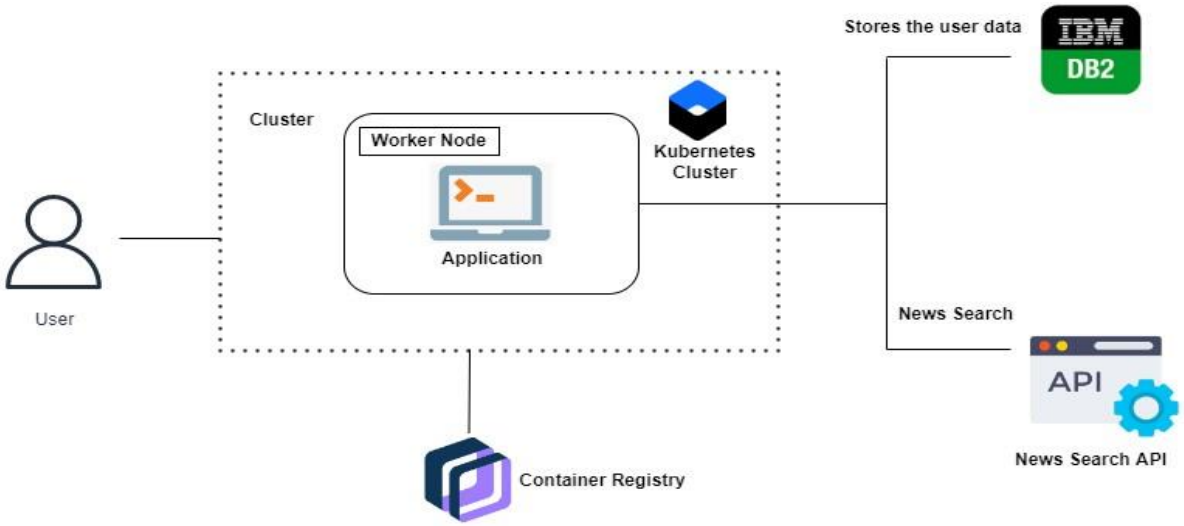


Fig 5.2: Solution architecture

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint planning is an event in scrum that kicks off the sprint. The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved. Sprint planning is done in collaboration with the whole scrum team. The sprint is a set period of time where all the work is done. However, before leap intoaction it is necessary to set up the sprint. It needs to decide on how long the time box is goingto be, the sprint goal, and where it is going to start. The sprint planning session kicks off the sprint by setting the agenda and focus. Sprint Planning and estimation for Inventory Management system for Retailers is shown in table 6.1.

Activity Number	Activity	Sub Activity	Assigned To	Status
1.	Setting up Application Environment	• Create Flask Project • Create IBM Cloud Account • Install IBM Cloud CLI • Docker CLI Installation • Create An Account In Sendgrid	All Members	Completed
2.	Implementing Web Application	• Create UI To Interact With Application	All Members	Completed
3.	Integrating SendGrid Service	• SendGrid Integration With Python Code	All Members	Completed
4.	Deployment of App InIBM Cloud	• Containerize The App • Upload Image To IBM Container Registry • Deploy in Kubernetes	All Members	Completed
5.	Ideation Phase	• Literature Survey On The Selected Project & Information Gathering • Prepare Empathy Map • Ideation	All Members	Completed
6.	Project Design Phase – I	• Proposed Solution • Problem Solution Fit • Solution Architecture	All Members	Completed

7.	Project PlanningPhase	• Prepare Milestone & Activity List • Sprint Delivery Plan	All Members	Completed
8.	Project DevelopmentPhase	• Delivery Of Sprint-1 • Delivery Of Sprint-2 • Delivery Of Sprint-3 • Delivery Of Sprint-4	All Members	Completed

Table 6.1: Sprint Planning

6.1 SPRINT DELIVERY SCHEDULE

The sprint delivery plan is scheduled accordingly as shown in the below table 6.2 which consists of the sprints with respective to their duration, sprint start and end date and the releasing data. Sprint Planning done for Inventory Management system for Retailers is shown in table 6.2.

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint 1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	5	High	
Sprint 1		USN-2	As a user, I will receive confirmation email onceI have registered for the application	4	High	All Members
Sprint 1		USN-3	As a user, I can register for the application through Gmail	3	Medium	All Members
Sprint 1	Login	USN-4	As a user, I can log into the application by entering email & password	4	High	All Members

Sprint 1	Dashboard	USN-5	As a user, I can see the latest news and headlines	4	High	All Members

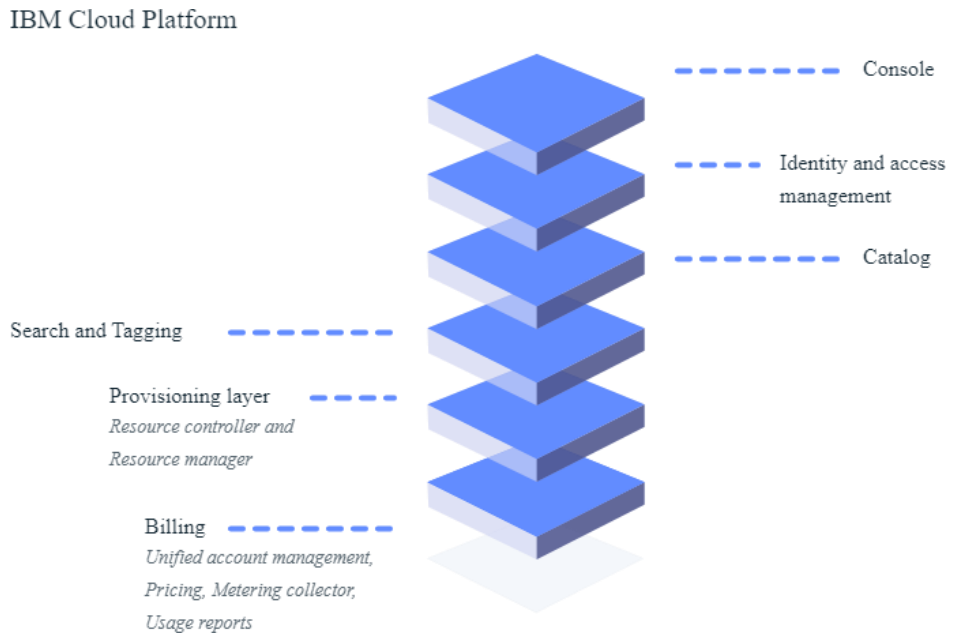
Table 6.2: Sprint Planning

CHAPTER 7
CODING & SOLUTIONING

IBM Cloud

The IBM Cloud platform combines platform as a service (PaaS) with infrastructure asa service (IaaS) to provide an integrated experience. The platform scales and supports both small development teams and organizations, and large enterprise businesses. Globally deployed across data centers around the world, the solution you build on IBM Cloud spins upfast and performs reliably in a tested and supported environment you can trust!

IBM Cloud provides solutions that enable higher levels of compliance, security, and management, with proven architecture patterns and methods for rapid delivery for running mission-critical workloads.



Flask framework

Flask is a micro web framework written in Python. It is classified asa microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add

application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

IBM DB2 Module

Module features allow you to

- Extend schema support by allowing you to group together, in a named set, a collectionof related data type definitions, database object definitions and other logic elements including:
    - SQL procedures
    - A module initialization procedure for implicit execution upon moduleinitialization
    - User-defined data type definitions including: distinct type, array type,associative array type, row type, and cursor type
- Define a namespace such that objects defined within the module can refer to otherobjects defined in the module without providing an explicit qualifier.
- Add object definitions that are private to the module. These objects can only bereferenced by other objects within the module.
- Add object definitions that are published. Published objects can be referenced fromwithin the module or from outside of the module.
- Define published prototypes of routines without routine-bodies in modules and later implement the routine-bodies using the routine prototype.
- Initialize the module by executing the module initialization procedure for the module.This procedure can include SQL statements, SQL PL statements, and can be used to set default values for global variables or to open cursors.
- Reference objects defined in the module from within the module and from outside ofthe module by using the module name as a qualifier (2-part name support) or a combination of the module name and schema name as qualifiers (3-part name support).
- Drop objects defined within the module.
- Drop the module.
- Manage who can reference objects in a module by allowing you to grant and revokethe EXECUTE privilege for the module.

Docker CLI

The Docker client enables users to interact with Docker. The Docker client can resideon the same host as the daemon or connect to a daemon on a remote host. A docker client can communicate with more than one daemon. The Docker client provides a command line interface (CLI) that allows you to issue build, run, and stop application commands to a Docker daemon.

The main purpose of the Docker Client is to provide a means to direct the pull of images froma registry and to have it run on a Docker host. Common commands issued by a client are:

- docker build
- docker pull
- docker run

IBM cloud CLI

IBM Cloud CLI provides full management of your IBM Cloud account via commandline. Some installation steps described along this guide may need the IBM Cloud Command Line Interface (CLI) available to be performed.

SendGrid API

SendGrid's web API allows users to pull information about their email program without having to actually log on to SendGrid.com. Users can pull lists, statistics, and evenemail reports. In addition to this, users can send email via the web API without
using traditional SMTP.

Kubernetes

Kubernetes is an open-source Container Management tool which automates container deployment, container scaling, and descaling and container load balancing (also called as container orchestration tool). It is written in Golang and has a huge community because it was first developed by Google and later donated to CNCF (Cloud Native Computing Foundation). Kubernetes can group 'n' number of containers into one logical unit for managing and deploying them easily. It works brilliantly with all cloud vendors i.e. public, hybrid and on- premises. Kubernetes is an open-source platform that manages Docker containers in the form of a cluster. Along with the automated deployment and scaling of containers, it provides healing by automatically restarting failed containers and rescheduling them when their hosts die. This capability improves the application's availability.

CHAPTER 8

TESTING AND RESULTS

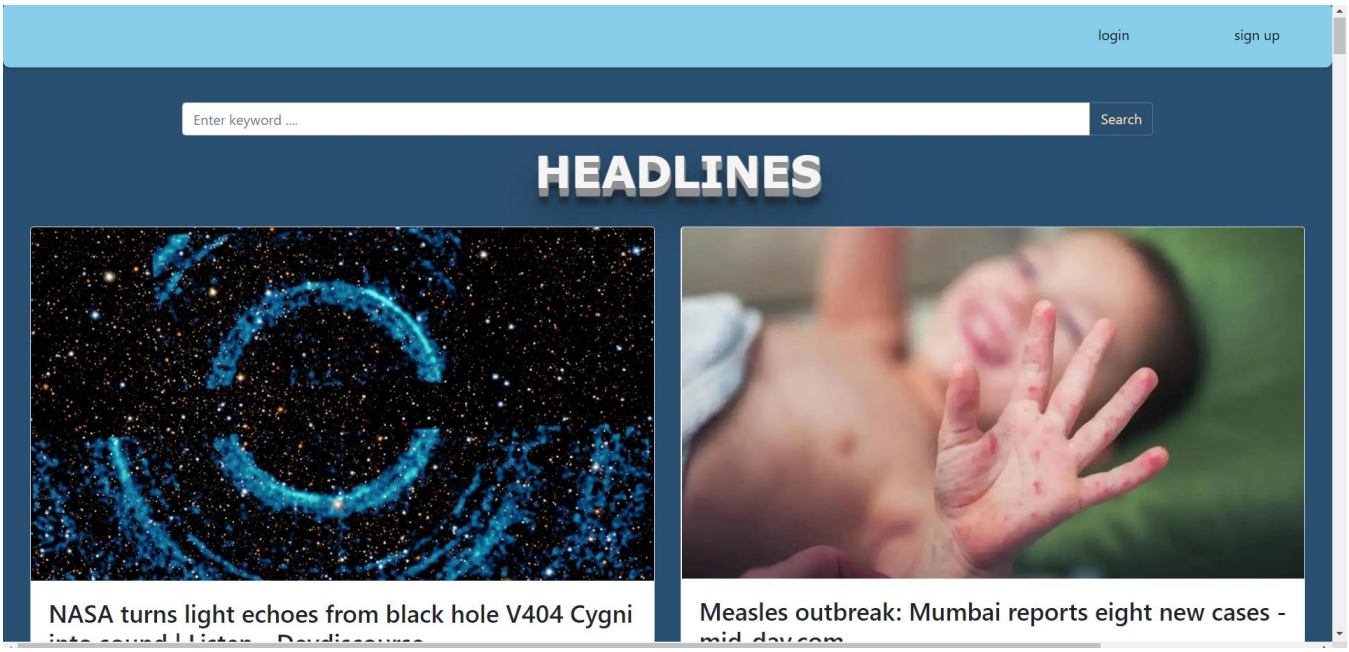


Fig 8.1 : Home page

Fig 8.2: Login Page

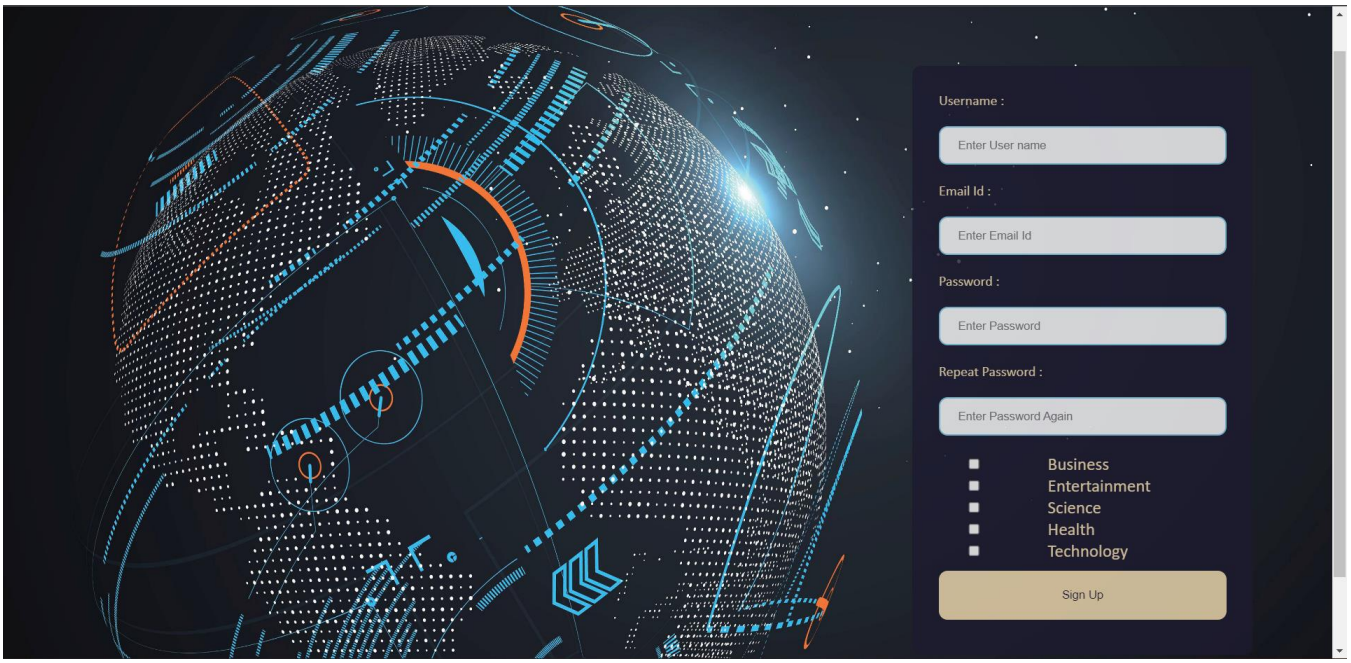


Fig 8.3: signin Page

CHAPTER 9

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- Viewers can get their news straight off their smartphone or tablet computer.
- News is at their fingertips in an instant. An online newspaper can be read more elaborate than a printed newspaper.
- You can read the old issues too very easily at the click of the mouse.
- This will help the users to share news on various platforms such as Twitter and Facebook. This will not only give an amazing user experience

DISADVANTAGES

- It can be limited by time.
- It may rely too heavily on personalities, emotions, opinions... not facts.
- It can shortchange complex stories or avoid them altogether.

CHAPTER 10

CONCLUSION

The way we consume news has shifted dramatically in the last decade and having a dedicated website is no longer enough. Users expect updates to be immediately available and accessible via multiple devices, and easy to share across their social media networks. News apps have also become increasingly important for users who want to avoid consuming news via social media and digest news from a reliable source.

CHAPTER 11
FUTURE SCOPE

In a nut shell, the future scope of the project can be summarized as follows,We can add more advanced software that includes more facilities.Implement backup mechanism so that the user's data can be restored.Create multiple load balancers to split up the load

.

SOURCE CODE

```
# import libraries
from flask import Flask, render_template, request,redirect,url_for
import jinja2
from newsapi.newsapi_client import NewsApiClient
import ibm_db
import pycountry

# init flask app
app = Flask(__name__)

print(".....Connecting to database.....")

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=2f3279a5-73d1-4859-88f0-
a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PROTOCOLO=TCPIP;PORT=30756;
Security=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=pbp31273;PWD=0mhmHAwaShM
BEWZw","","")

print("_____connected_____")

# Init news api
newsapi = NewsApiClient(api_key='0ba9ccf85f044894877b98fcf7bb669c')

# helper function
def get_sources_and_domains():
    all_sources = newsapi.get_sources()['sources']
    sources = []
    domains = []
    for e in all_sources:
        id = e['id']
        domain = e['url'].replace("http://", "")
        domain = domain.replace("https://", "")
        domain = domain.replace("www.", "")
        slash = domain.find('/')
        if slash != -1:
            domain = domain[:slash]
        sources.append(id)
        domains.append(domain)
    sources = ", ".join(sources)
    domains = ", ".join(domains)
    return sources, domains

@app.route("/", methods=['GET', 'POST'])
@app.route('/home', methods=['GET', 'POST'])
def home():
    if request.method == "POST":
        sources, domains = get_sources_and_domains()
        keyword = request.form["keyword"]
```

```
        related_news =
newsapi.get_everything(q=keyword,sources=sources,domains=domains,language='en',sort_by=
'relevancy')
        no_of_articles = related_news['totalResults']
        if no_of_articles > 100:
            no_of_articles = 100
        all_articles =
newsapi.get_everything(q=keyword,sources=sources,domains=domains,language='en',sort_by=
'relevancy',page_size = no_of_articles)['articles']
        return render_template("home.html", all_articles =
all_articles,keyword=keyword)
    else:
        top_headlines = newsapi.get_top_headlines(country="in", language="en")
        total_results = top_headlines['totalResults']
        if total_results > 100:
            total_results = 100
        all_headlines =
newsapi.get_top_headlines(country="in",language="en",page_size=total_results)['articles
']
        return render_template("home.html", all_headlines = all_headlines)
    return render_template("home.html")


@app.route("/user", methods=['GET', 'POST'])
def user():
    if request.method == "POST":
        sources, domains = get_sources_and_domains()
        keyword = request.form["keyword"]
        related_news =
newsapi.get_everything(q=keyword,sources=sources,domains=domains,language='en',sort_by=
'relevancy')
        no_of_articles = related_news['totalResults']
        if no_of_articles > 100:
            no_of_articles = 100
        all_articles =
newsapi.get_everything(q=keyword,sources=sources,domains=domains,language='en',sort_by=
'relevancy',page_size = no_of_articles)['articles']
        return render_template("home.html", all_articles =
all_articles,keyword=keyword)
    else:
        c1 = request.args.get('c1')
        c2 = request.args.get('c2')
        c3 = request.args.get('c3')
        c4 = request.args.get('c4')
        c5 = request.args.get('c5')
        options = [c1, c2, c3, c4, c5]
        option = []
        all_headlines = []
        for ele in options:
            if ele != 'NULL':
```

```
                option.append(ele)
        print("USERS CHOICES : ", option)
        input_country = "India"
        input_countries = [f'{input_country.strip()}']
        countries = {}
        for country in pycountry.countries:
            countries[country.name] = country.alpha_2
            codes = [countries.get(country.title(), 'Unknown code') for country in
input_countries]
        for ele in option:
            top_headlines = newsapi.get_top_headlines(category=f'{ele.lower()}',
language='en', country=f'{codes[0].lower()}')
            for val in top_headlines['articles']:
                all_headlines.append(val)
        return render_template("user.html", all_headlines = all_headlines)
    return render_template("user.html")

@app.route("/login", methods=["GET", "POST"])
def login():
    if request.method == 'GET':
        return render_template('login.html')
    if request.method == 'POST':
        user_name = request.form.get('username')
        password = request.form.get('password')
        query = ibm_db.exec_immediate(conn,"SELECT UNAME,PASSWORD,C1,C2,C3,C4,C5 FROM
USERS WHERE UNAME='"+user_name+"'")
        res = ibm_db.fetch_both(query)

        r = {}
        i = 0
        for ele in res:
            if i % 2 == 0:
                r[ele] = res[ele].strip('+ ')
            i += 1
        print("results", r)
        if res:
            if res["PASSWORD"].strip() == password:
                print(password, user_name)
                return redirect(url_for('user', c1 = r['C1'], c2 = r['C2'], c3 =
r['C3'], c4 = r['C4'], c5 = r['C5']))
        return redirect(url_for('login'))

@app.route("/Registration", methods=["GET", "POST"])
def Registration():
    if request.method == "GET":
        return render_template('registration.html')
    if request.method == 'POST':
        user_name = request.form.get('username')
        password = request.form.get('password')
        email = request.form.get('email_id')
```

```
        print(request.form)
        choice_list = [request.form.get('C1'), request.form.get('C2'),
request.form.get('C3'), request.form.get('C4'), request.form.get('C5')]
        for i in range(len(choice_list)):
            if choice_list[i] == None:
                choice_list[i] = 'NULL'
        print(choice_list)
        ibm_db.exec_immediate(conn, "INSERT INTO USERS VALUES
('"+user_name+"','"+password+"','"+email+"', '"+str(choice_list[0])+"',
'"+str(choice_list[1])+"', '"+str(choice_list[2])+"', '"+str(choice_list[3])+"',
'"+str(choice_list[4])+"')")
        return redirect(url_for('login'))

if __name__ == "__main__":
    app.run(debug = True, host='0.0.0.0')
```

GITHUB LINK:

IBM-EPBL/IBM-Project-54925-1663171176: News Tracker Application (github.com)

DRIVE LINK:

https://drive.google.com/file/d/14yhPtqRgK-xmkgCLYqUIuJz61aWnjWLi/view?usp=drivesdk

REFERENCES

https://www.researchgate.net/publication/299870645_Exploring_mobile_news_reading_interactions_for_news_app_personalisation

https://www.researchgate.net/publication/227438253_Following_the_Fed_with_a_News_Tracker

https://www.researchgate.net/publication/361087328_An_End-to-end_Weakly-supervised_News_Aggregation_Framework