**A**

**PROJECT REPORT**

**ON**

**SAMACHAR INDIA**

THE NEWS PORTEL

**Submitted Towards Partial Fulfillment of**

**Three-Year diploma in**

# Computer Science& Engineering

**SUBMITTED BY: SUBMITTED TO:**

**ADITYA PRAKASH DUBEY VIVEK SRIVASTAVA(HOD)**

**Session 2022-2023**



The system development was organized into 5 major parts:

1. Requirement Gathering
2. Documentation/Design
3. Development
4. Coding
5. Testing

**DECLARATION**

This is to certify that the project report entitled “**SAMACHAR INDIA**” is done by me is an authentic work carried out for the partial fulfillment of the requirements for the award of the Diploma in **“(Computer Science And Engineering)”** . The matter embodied in this project work has not been submitted earlier for award of any degree or diploma to the best of my knowledge and belief.

**ADITYA PRAKASH DUBEY (E20480935500034)**

**INDEX**

1. **Introduction………………………………………………………**
   1. 1.1 Introduction……………………………………………………………
   2. 1.2 Objectives………………………………………………………………
   3. 1.3 Problem Definition……………………………………………………

**2. System Analysis……………………………………………………**

* 1. 2.1 Objective……………………………………………………………
  2. 2.2 SDLC Phases…………………………………………………………
  3. 2.2.1 Feasibility Study………………………………………………
  4. 2.2.2 Report Approval………………………………………………
  5. 2.2.3 System Analysis…………………………………………………
  6. 2.2.4 System Design……………………………………………………
  7. 2.2.5 Coding……………………………………………………………
  8. 2.2.6 Testing…………………………………………………………
  9. 2.2.7 Implementation…………………………………………………
  10. 2.2.8 Maintenance………………………………………………………
  11. 2.3 Process Description……………………………………………………
  12. 2.4 Project Model Used…………………………………………………
  13. 2.5 ER-Diagram……………………………………………………………

2.6 Data Flow Diagram……………………………………………………

**3. Software Requirement Specification………….……………**

* 1. 3.1 Hardware Requirement…………………………………………………
  2. 3.2 Software Requirement………………………………………………….
  3. 3.3 Support Maintenance…………………………………………………….
  4. **4. System Design Approach…………………………………………**
  5. 4.1 Top-Down Designing………………………………………………
  6. 4.2 Bottom –Up Designing……………………………………………….
  7. 4.3 Following Approach…………………………………………………
  8. **5. Backend Design…………………………………………….................**
  9. 5.1 Description of Classes and Methods (model.py) ………………
  10. 5.2 Defined urls (urls.py)……………………………………………

1. **Data Modeling…………………………………………………**

6.1 List of Tables……………………………………………………

6.2 Structure of Tables…………………………………………………

* 1. **7. Testing…………………………………………………………**
  2. **8. Input-Output Forms……………………………………………**
  3. 8.1 Project Screenshot………………………………………………………
  4. 8.2 Project Coding…………………………………………………………….
  5. **9. Future Scope………………………………………………………**
  6. **10. Conclusion……………………………………………………….**

**LIST OF TABLES**

1. Group Table…………………………………………………
2. User Table…………………………………………………..
3. City Table..…………………………………………………..
4. Contact us Table ……………………………………………
5. IGallery Table ……………………………………………..
6. MyNews Table …………………………………………….
7. NCategory Table ………………………………………….
8. Slider Table ………………………………………………
9. VideoNews Table …………………………………………..

**LIST OF FIGURES**

1.Level 0 Data Flow Diagram………………………………………………………….



**1.1 PROJECT INTRODUCTION**

Newspaper cannot be defined by the secondword-paper. They’ve got to be defined bythe .Firs***t*** *word-news. It provides de*tail information about the issues taking place all over the world.

Samachar India never works on the rumors. It transfers the real news to the people. If you don’t read the newspaper you’re uninformed. If you read the newspaper, you’re misinformed.

**1.3 OBJECTIVES**

* It highlights the issues taking place most frequently.
* Provides real news to the people.
* Works for the people and by the people.
* Its main objective is to grasp people’s attention and convince them with the information provided.
* It believes that “**More information is always better than less.”**
* It provides news with humor, terseness, descriptive, good literary.

**1.4 PROBLEM DEFINITION**

In this section we shall discuss the limitation and drawback of the existing system that forced us to take up this project. Really that work was very typical to manage the daily errors free records and adding or removing any node from server. This problem produces a need to change the existing system. Some of these shortcomings are being discussed below: -

* **Low Functionality**

With the existing system, the biggest problem was the low functionality. The problem faced hampered the work. For small task like adding any new node to server or deleting a node or keeping daily record we have to appoint minimum two or three employee.

* **Erroneous Input and Output**

In the existing system, humans performed all the tasks. As in the human tendency, error is also a possibility. Therefore, the inputs entered by the person who is working in the Company, in the registers may not be absolutely foolproof and may be erroneous. As a result of wrong input, the output reports etc. Will also be wrong which would in turn affect the performance.

* **Portability Problem**

System that existed previously was manual. As a result, the system was less portable. One has to carry the loads of many registers to take the data from one place to another. A big problem was that the system was less flexible and if we wanted to calculate yearly or monthly maintenance report or efficiency report, then it was a big headache.

* **Security-**

Security concerns were also one of the motives of the Company for the need of software. In the registers, the data is not secure as anybody can tamper with the data written in the registers. While in this software, just a password makes it absolutely secure from the reach of unauthorized persons.

* **Data Redundancy**

In the case of manual system, the registers are maintained in which, a lot of data is written

.

* **Processing Speed**

In manual system maintaining a register and performing the necessary calculation has proved to be a troublesome job, which takes a lot of time and may affect the performance of the Company. But with this software we can have all the tasks performed in a fraction of second by a single click thus making the troublesome job much easier.

* **Manual Errors**

When a number of tough tasks are prepared by the humans like preparation of reports, performing long calculation then some human errors are obvious due to a number of factors like mental strain, tiredness etc. But as we all know that computer never get tired irrespective of the amount of work it has to do. So, this software can nullify the probability of manual error that improve the performance.

* **Complexity in Work**

In manual system whenever a record is to be updated or to be deleted a lot of cutting and overwriting needs to be done on the registers that are concerned that are deleted or updated record, which makes the work very complex.

1. **SYSTEM ANALYSIS**

**2.1 Objective:**

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components.

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

**Analysis specifies what the system should do.**

**2.2 SDLC Phases:**

System Development Life Cycle (SDLC) mainly consists of the following 7 phases which can be detailed: -

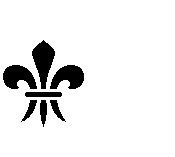
**2.2.1 Preliminary Investigation: -**

This is the first phase of the system development life cycle. In this phase we tend to find out the needs of the client –what exactly does the client want? Before the development of any system the important point is to know the needs, objectives and scope of the system

**2.2.1 Feasibility Study**: -

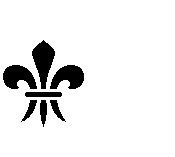
Feasibility study is the preliminary investigation step of the system development life cycle. Things are always easy at the beginning in any software process. In fact, nothing is in feasible with unlimited time and resources. But it is not the fact. So, practically we have to do in limited resources in a restricted time margin. So for the system to be feasible, following points we have to consider.

The feasibility study is conducted to check whether the candidate system is feasible. The system which is selected to be the best against the criteria is there after designed and developed. The feasibility study takes in to consideration, the risks involved in the project development beforehand. Therefore, in this phase we have to do feasibility study which is the test of the website according to its work ability, impact on the organization, ability to meet user need and effective use of resources. We do the feasibility study for website to analyze the risks, costs and benefits relating to economics, technology and user organization. There are several types of feasibility depending on the aspect they cover. Import of these includes:

** Technical Feasibility:**

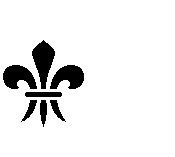
This is an important outcome of preliminary investigation. It comprises of following questions:

* Can the work of project bed one with the current equipment, existing software and available man power resource?
* If Technology is required what are the possibilities that it can be developed?

 **Economic Feasibility:**

It deals with question related to the economy. It comprises of the following questions: -

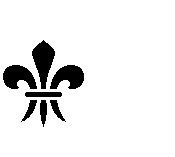
* + Are there sufficient benefits in creating the system to make the cost acceptable?
  + Are the costs of not creating the system so great that the project must be undertaken?

 **Legal Feasibility:**

It deals with the question related to the legal issues. It comprises of the following

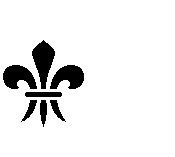
questions: -

* Contract Signing
* Software License agreement
* Issues related to cyber laws.
* Legal issues relating to the man power contract.

 **Operational Feasibility:**

The operational feasibility consists of the following activity: -

* Will the system be useful if it is developed &implemented?
* Will there be resistance from employee?

**Social & Behavioral Feasibility:**

It deals with the various issues related to the human behavior like: -

* Whether the user be able to adapt a new change or not?
* Whether the ambiance we are providing suits the user or not?

**2.2.2 Report Approval: -**

Request approval is the preliminary investigation phase of system development lifecycle. Request approval is the phase in which all the requirements which would be provide in the system are stated. The request approval is a sort of agreement between the client and the company which is building this software. Both the parties should be mutually agreed on the stated requirements.

**2.2.3 System Analysis:** -

System analysis is the phase following the phase of the request approval. In this phase we tend to analyze the overall system which we have to build. System analysis is the crucial part in SDLC.

**2.2.4 System Design:** -

System design means the designing of the system. The System can be done in either of the following two ways: -

* + Logical System Design
  + Physical System Design

**2.2.5 Coding:** -

Coding is the phase in which a developer codes using any programming languages. Coding constitutes only 20 % of the whole project and which is easier to write. The coding work is also done in the teams; development of the system is usually done under the modular programming style, which can be either top-down approach or bottom-up approach.

**2.2.6 Testing:** -

Testing is the phase in which the system that has been developed is tested. Testing comprises of the 60%of the overall development of the system. Testing of the system is important because testing aims to uncover the different errors in the system. There are various different testing techniques that can be used for the testing of the system.

**2.2.7 Implementation: -**

Implementation process involved the installation of software on user’s side. Implementation process actually depends on type of a system & various. Opting for suitable conversion approach is a step implementation. The conversion processes are as follows: -

* + - Parallel Conversion
    - Direct Conversion Approach
    - Pilot Conversion Approach
    - Phase In Conversion Approach

**2.2.8 Maintenance**: -

Merely developing the system is not important but also maintenance is important. The company that has built the system provides for some time free of cost maintenance to the client and after that period it is usually a paid service.

s

**2.3 Process Description**

Gantt charts mainly used to allocate resources to activities. The resources allocated to activities include staff, hardware, and software. Gantt charts (named after its developer Henry Gantt) are useful for resource planning. A Gantt chart is special type of bar chart where each bar represents an activity. The bars are drawn along a timeline. The length of each bar is proportional to the duration of the time planned for the corresponding activity.

Gantt chart is a project scheduling technique. Progress can be represented easily in a Gantt chart, by coloring each milestone when completed. The project will start in the month of January and end after 4 months at the beginning of April.

**2.4 PROJECT MODEL USED**

**Iterative Enhancement Model**

* This model has the same phases as the waterfall model, but with fewer restrictions. Generally, the phases occur in the same order as in the waterfall model, but they may be conducted in several cycles.
* Useable product is released at the end of the each cycle, with each release providing additional functionality. Customers and developers specify as many requirements as possible and prepare a SRS document. Developers and customers then prioritize these requirements. Developers implement the specified requirements in one or more cycles of design, implementation and test based on the defined priorities.

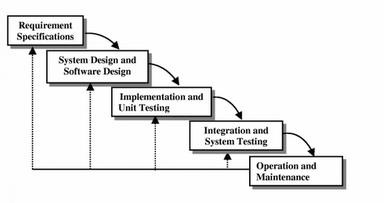
The procedure itself consists of the initialization step, the iteration step, and the Project Control List. The initialization step creates a base version of the system. The goal for this initial implementation is to create a product to which the user can react. It should offer a sampling of the key aspects of the problem and provide a solution that is simple enough to understand and implement easily. To guide the iteration process, a project control list is created that contains a record of all tasks that need to be performed. It includes such items as new features to be implemented and areas of redesign of the existing solution. The control list is constantly being revised as a result of the analysis phase.

The iteration involves the redesign and implementation of iteration is to be simple, straightforward, and modular, supporting redesign at that stage or as a task added to the project control list. The level of design detail is not dictated by the iterative approach. In a light-weight iterative project the code may represent the major source of [documentation](http://en.wikipedia.org/wiki/Software_documentation) of the system; however, in a critical iterative project a formal [Software Design Document](http://en.wikipedia.org/wiki/Software_Design_Document) may be used. The analysis of an iteration is based upon user feedback, and the program analysis facilities available. It involves analysis of the structure, modularity, [usability](http://en.wikipedia.org/wiki/Usability), reliability, efficiency, & achievement of goals. The project control list is modified in light of the analysis results.

**PHASES:**

Incremental development slices the system functionality into increments (portions). In each increment, a slice of functionality is delivered through cross- discipline work, from the requirements to the deployment. The unified process groups increments/iterations into phases: inception, elaboration, construction, and transition.

* Inception identifies project scope, requirements (functional and non-functional) and risks at a high level but in enough detail that work can be estimated.
* Elaboration delivers a working architecture that mitigates the top risks and fulfills the non-functional requirements.
* Construction incrementally fills-in the architecture with production-ready code produced from analysis, design, implementation, and testing of the functional requirements.
* Transition delivers the system into the production operating environment.



**2.5 ER-Diagram**

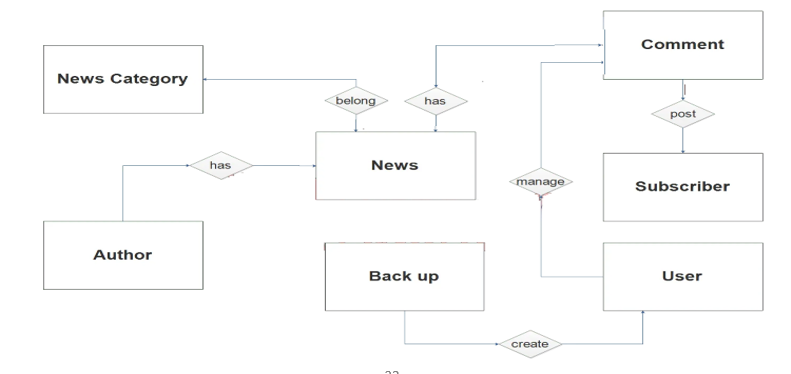
**Introduction: -**

I[n software engineering, an](http://en.wikipedia.org/wiki/Software_engineering) entity-relationship model (ERM) is an abstract and conceptual representation of [data.](http://en.wikipedia.org/wiki/Data) Entity-relationship modeling is a [database modeling](http://en.wikipedia.org/wiki/Database_model) [method,](http://en.wikipedia.org/wiki/Database_model) used to produce a type [of conceptual schema](http://en.wikipedia.org/wiki/Conceptual_schema) o[r semantic data model of](http://en.wikipedia.org/wiki/Semantic_data_model) a system, often [a relational database, a](http://en.wikipedia.org/wiki/Relational_database)nd its require[ments in a top-down](http://en.wikipedia.org/wiki/Top-down) fashion. Diagrams created by this process are called entity-relationship diagrams, ER diagrams, or ERDs. ER Diagrams depicts relationship between data objects. The attribute of each data objects noted in the entity-relationship diagram can be described using a data object description. Entity relationship diagram is very basic, conceptual model of data and it is fundamental to the physical database design. This analysis is then used to organize data as relations, normalizing relations, and obtaining a Relational database.

The entity-relationship model for data uses three features to describe data. These are:

1. Entities which specify distinct real-world items in an application.
2. Relationship, which connect entities and represent meaningful dependencies between them.
3. Attributes which specify properties of entities &relationships.

**E-R Diagram**

****

**2.6 Data Flow Diagram**

**Introduction:**

DFD is an acronym for the word Data Flow Diagram. DFD is ppictorial representation of the system DFD is a graphical representations of the flow of data through the information system. DFD are also used for the visualization of data processing (structured design). ADFD provides no information about the timings of the process, or about whether process will operate in parallel or sequence. DFD is an important technique for modeling system’s high-level detail by showing how input data is transformed to output results through a sequence of functional transformations. DFD reveal relationships among between the various components in a program or system. The strength of DFD lies in the fact that using few symbols we are able to express program design in an easier manner. ADFD can be used to represent the following:

* External Entity sending and receiving data. Process that changes the data.
* Flow of data within the system. Data Storage locations.

**Uses of DFD:**

The main uses of data flow diagrams are as follows: -

DFD is a method of choice for representation of showing of information through a system because of the following reasons:

* + - DFDs are easier to understand by technical and non-technical audiences.
    - DFDs can provide a high- level system overview, complete with boundaries and connections to other system.
    - DFDs can provide a detailed representation of system components.

**0 level DFD**

News Type

Management

News

Management

Articles

Management

News

System

Management

Content

Management

Login Management

**1 Level DFD**

News

Management

Generate

News Report

Articles

Management

Generate

Articles Report

Content

Management

Generate

Content Report

System User

Management

Login

Management

News Type

Management

Generate News

Type Report

Generate System

User Report

Check user

Login details

**3. SOFTWARE REQUIREMENT SPECIFICATION**

A requirements specification for a software system is a complete description of the behavior of a system to be developed and it includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements.

Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints) Requirements are a sub-field of software engineering that deals with the elicitation, analysis, specification, and validation of requirements for software.

The software requirement specification document enlists all necessary requirements for project development. To derive the requirements, we need to have clear and thorough understanding of the products to be developed. This is prepared after detailed communications with project team and the customer

**3.1 SERVER SIDE HARDWARE REQUIREMENT:**

 AMD Athlon 64 with processor speed 2.8 or more

 256 DDR Ram

 40 GB Hard disk

 Network Interface card

 IIS

 CD-Drive

**3.2 SERVER SIDE SOFTWARE REQUIREMENT:**



 Windows

 Python 3.7 and PyCharm IDE 2021.3

 SQLite

**CLIENT SIDE HARDWARE REQUIREMENT:**



 Processor Dual core based computer

 2 GB Minimum RAM

 20 GB HDD

 100 Mbps LAN

 Web Browser

**TO DEVELOP THIS PROJECT THE VARIOUS SOFTWARE RESOURCES ARE USED.**

* Front End               -  HTML-CSS & Bootstrap
* Back End                -  SQLite
* Web Server             -  Apache Server
* Technology               - Python technology
* Code-Behind Language    - Python
* IDE                                 - PyCharm
* Data Base Server: MYSQL Database, Operating System (windows Server).

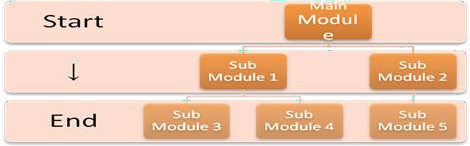
**3.3 SUPPORT AND MAINTENANCE:-**

One-year free support for rectifying system bugs including front end and beck end will be provided. During warranty period Software Engineers will be responsible for removing bugs and improving it. After one year support can be extended @ 20% of the total product deployment cost.

**4. SYSTEM DESIGN APPROACH**

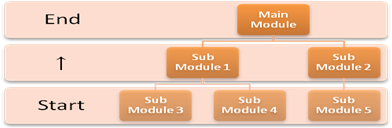
**4.1 Top – Down designing:**

The top - down designing approach started with major components of the system. It is a stepwise refinement which starts from an abstract design, in each steps the design is refined two or more concrete levels until we reach a level where no – more refinement is possible or not needed.



**4.2 Bottom – Up designing:**

In bottom – up designing the most basic and primitive components are designed first, and we proceed to higher level components. We work with layers of abstractions and abstraction are implemented until the stage is reached where the operations supported by the layer is complete.



**4.3 Following Approach:**

In this project we are following **Mixed Approach** i.e. A combination of top – down and bottom – up. We are developing some of the components using top – down designing approach (e.g.the Web Pages) and the some components in bottom – up designing approach (e.g. the middle tier classes).

**5. BACKEND DESIGN**

**5.1 Description of Classes and Methods (model.py)**

**All code of views.py**

from django.shortcuts import render  
from .models import \*  
  
  
# Create your views here.  
def index(request):  
 data=slider.objects.all().order\_by("-id")[0:2]  
 d=ncategory.objects.all().order\_by('-id')[0:12]  
 x=mynews.objects.all().order\_by('-id')[0:8]  
 vdo = videonews.objects.all().order\_by('-id')[0:6]  
 ct=city.objects.all().order\_by('-id')[0:12]  
 mydict={'res':data,'data':d,"ndata":x,"vdata":vdo,"city":ct,}  
 return render(request,'user/index.html',mydict)  
  
##############################################  
  
def about(request):  
 return render(request,'user/about.html')  
##############################################  
  
def contact(request):  
 status=False  
  
 if request.method=="POST":  
 x=request.POST.get('name')  
 y=request.POST.get('email')  
 z=request.POST.get('mob')  
 a=request.POST.get('msg')  
 contactus(Name=x,Email=y,Mobile=z,Message=a).save()  
 status=True  
  
 return render(request,'user/contact.html',context={"msg":status})  
##############################################  
  
def video(request):  
 from django.db.models import Q  
 vdata=videonews.objects.all().order\_by('-id')  
 if request.method=="GET":  
 s=request.GET.get('search')  
 if s is not None:  
 vdata=videonews.objects.all().filter(Q(vhead\_\_icontains=s) | Q(vcategory\_\_icontains=s) |Q (vcity\_\_icontains=s))  
  
 md={"data":vdata}  
 return render(request,'user/video.html',md)  
  
  
##############################################  
def gallery(request):  
 d=igallery.objects.all().order\_by('-id')  
 mydict={"data":d}  
 return render(request,'user/gallery.html',context=mydict)  
##############################################  
  
def news(request):  
 x=ncategory.objects.all().order\_by('-id')  
 y=city.objects.all().order\_by('-id')  
 z=mynews.objects.all().order\_by('-id')  
 ctid=request.GET.get('ctid')  
 catid=request.GET.get('catid')  
 if ctid is not None:  
 z=mynews.objects.all().filter(ncity=ctid)  
 elif catid is not None:  
 z = mynews.objects.all().filter(category=catid)  
  
 mydict = {"cat": x, "ct": y, "ndata": z}  
  
 return render(request,'user/news.html',mydict)  
  
  
  
##############################################  
def ndetails(request):  
 nid=request.GET.get('n')  
 x=mynews.objects.all().filter(id=nid)  
 myd={"ndata":x}  
 return render (request,'user/ndetails.html',myd)  
  
  
########################### ####################  
def login(request):  
 return render(request,'user/login.html')  
  
  
  
def viewnews(request):  
 a=request.GET.get('msg')  
 c=request.GET.get('cid')  
 x=""  
 if a is not None:  
 x=mynews.objects.all().filter(category=a).order\_by('-id')  
 elif c is not None:  
 x=mynews.objects.all().filter(ncity=c).order\_by('-id')  
 mydict={"ndata":x}  
 return render(request,'user/viewnews.html',mydict)  
  
def vdetail(request):  
 a=request.GET.get('vid')  
 x=videonews.objects.all().filter(id=a)  
 md={"vdata":x}  
 return render(request,'user/vdetail.html',md)  
  
  
def profile(request):  
  
 return render(request,'user/profile.html')

**model.py**

rom django.db import models  
from tinymce.models import HTMLField  
  
# Create your models here.  
  
class contactus(models.Model):  
 Name=models.CharField(max\_length=100)  
 Email=models.CharField(max\_length=40)  
 Mobile=models.CharField(max\_length=30)  
 Message=models.TextField()  
 def \_\_str\_\_(self):  
 return self.Name  
  
class slider(models.Model):  
 shead=models.CharField(max\_length=300)  
 ssubject=models.CharField(max\_length=300)  
 sdes=models.TextField()  
 spic=models.ImageField(upload\_to='static/slider/',default="")  
 def \_\_str\_\_(self):  
 return self.shead  
######################################################################  
class igallery(models.Model):  
 gname=models.CharField(max\_length=400)  
 gpic=models.ImageField(upload\_to='static/gallery/',default="")  
 def \_\_str\_\_(self):  
 return self.gname  
  
#######################################################################  
class ncategory(models.Model):  
 ncategory=models.CharField(max\_length=80)  
 cpic=models.ImageField(upload\_to='static/category',null=True)  
 cdate=models.DateField()  
 def \_\_str\_\_(self):  
 return self.ncategory  
  
#########################################################################  
class city(models.Model):  
 ncity=models.CharField(max\_length=30)  
 cpic=models.ImageField(upload\_to='static/city')  
 cdate=models.DateField()  
 def \_\_str\_\_(self):  
 return self.ncity  
##########################################################################  
  
  
class mynews(models.Model):  
 ntitle=models.CharField(max\_length=500)  
 nhead=models.CharField(max\_length=500)  
 ndes=models.TextField()  
 npic=models.ImageField(upload\_to='static/news',null=True)  
 ncity=models.ForeignKey(city,on\_delete=models.CASCADE,null=True)  
 category=models.ForeignKey(ncategory,on\_delete=models.CASCADE,null=True)  
 ndate=models.DateField()  
  
class videonews(models.Model):  
 vlink=models.CharField(max\_length=200)  
 vhead=models.CharField(max\_length=500)  
 vcategory=models.CharField(max\_length=500)  
 vcity=models.CharField(max\_length=30)  
 vdate=models.DateField()  
 vdes=HTMLField()

**5.2 Defined urls (urls.py)**

**Project Level – URL.PY**

from django.contrib import admin  
from django.urls import path,include  
  
urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('user/',include('user.urls')),  
 path('',include('user.urls')),  
]

**APP LEVEL- URLS.PY**

from django.urls import path  
from . import views  
urlpatterns=[  
 path('index/',views.index),  
 path('',views.index),  
 path('home/',views.index),  
 path('about/',views.about),  
 path('contact/',views.contact),  
 path('gallery/',views.gallery),  
 path('video/',views.video),  
 path('news/',views.news),  
 path('login/',views.login),  
 path('viewnews/',views.viewnews),  
 path('details/',views.ndetails),  
 path('vdetail/',views.vdetail),  
 path('myprofile/',views.profile),  
  
  
  
  
]

**6. DATA MODELING**

**6.1 LIST OF TABLES:**

**6.1.1 Group Table**

**6.1.2 User Table**

**6.1.3 City Table**

**6.1.4 Contact us Table**

**6.1.5 IGallery Table**

**6.1.6 MyNews Table**

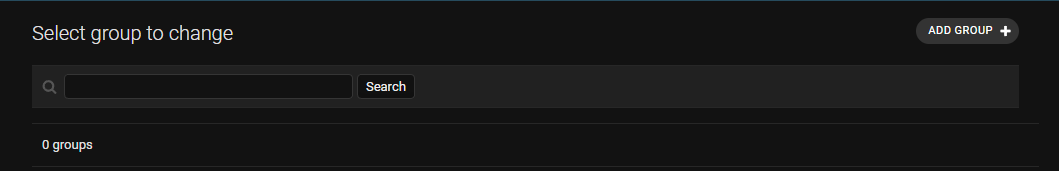
**6.1.7 NCategory Table**

**6.1.8 Slider Table**

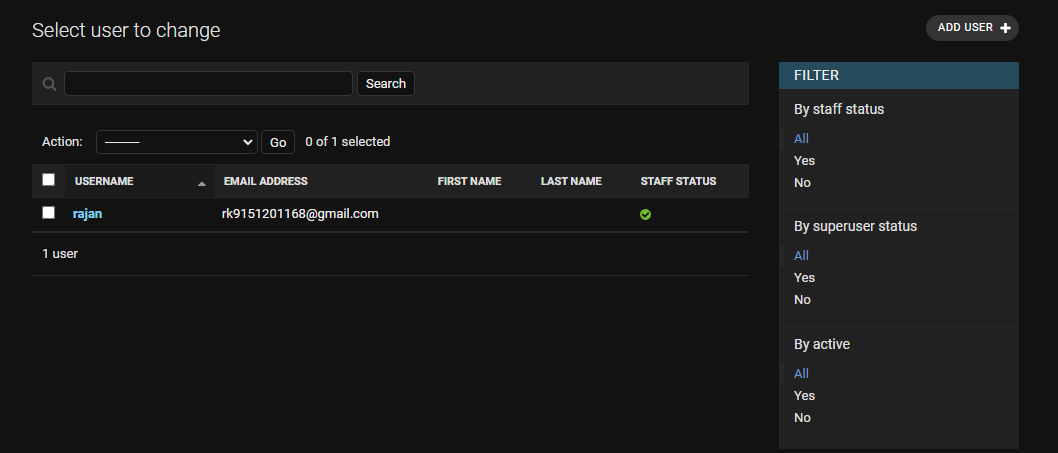
**6.1.9 VideoNews Table**

**6.2 STRUCTURE OF TABLES:**

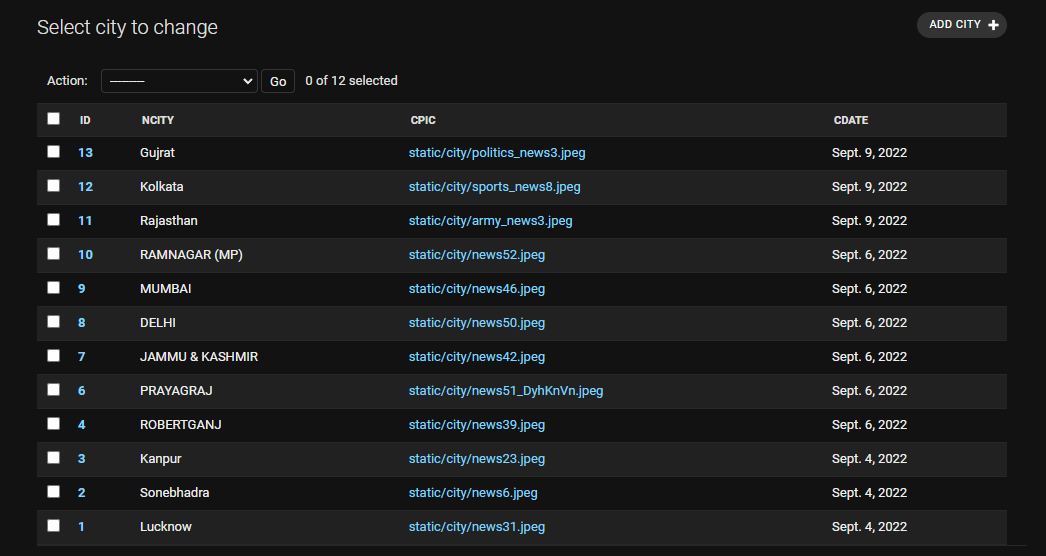
**6.2.1 Group Table**



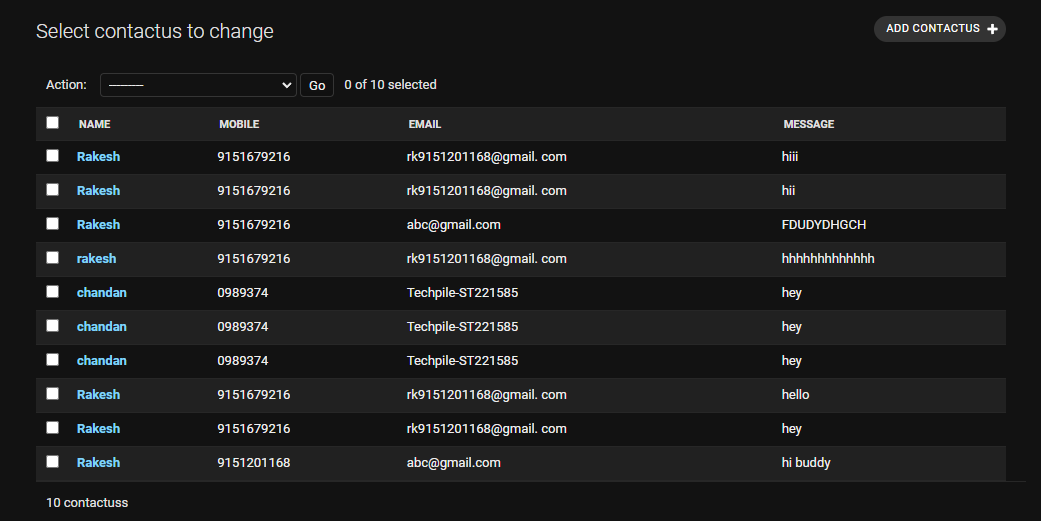
**6.2.2 User Table**



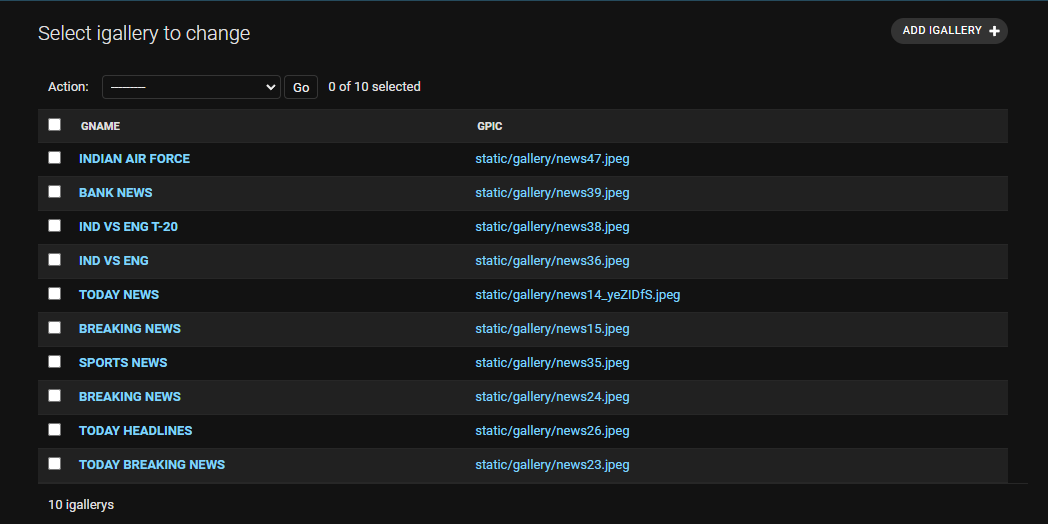
**6.2.3 City Table**



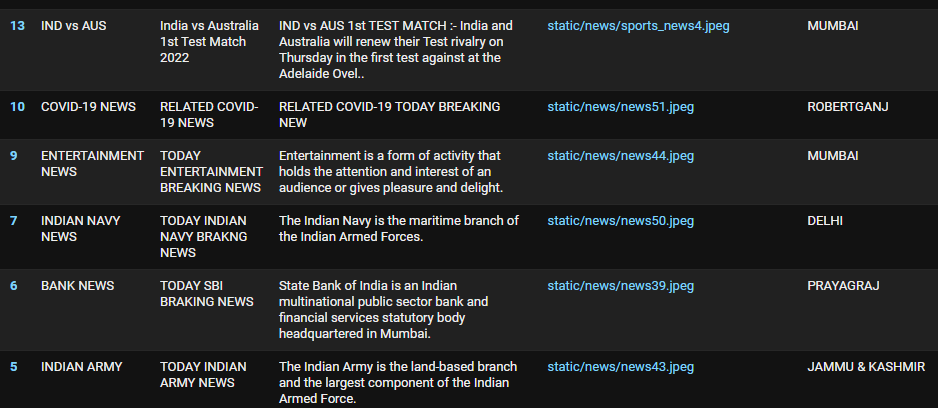
**6.2.4 Contact us Table**

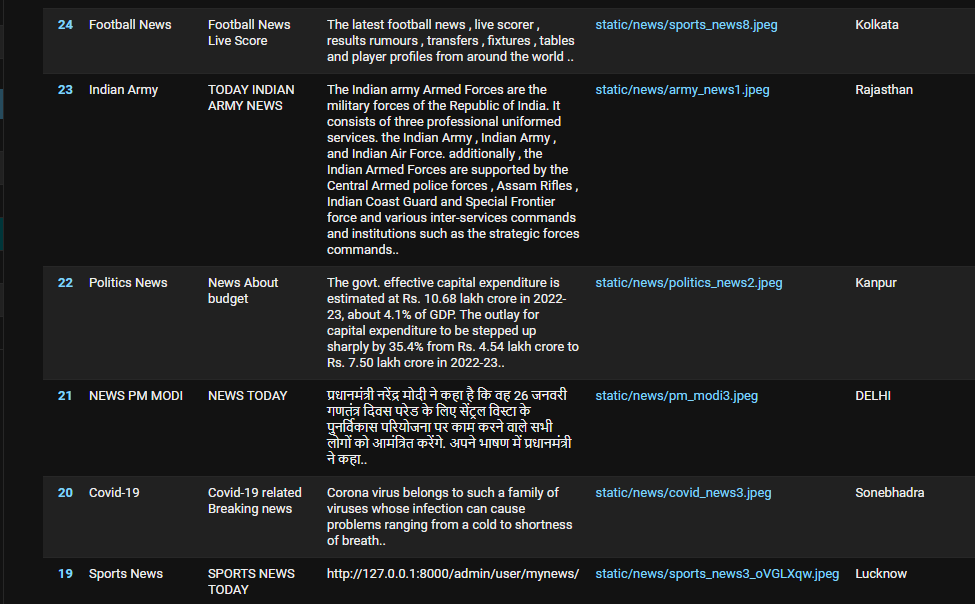


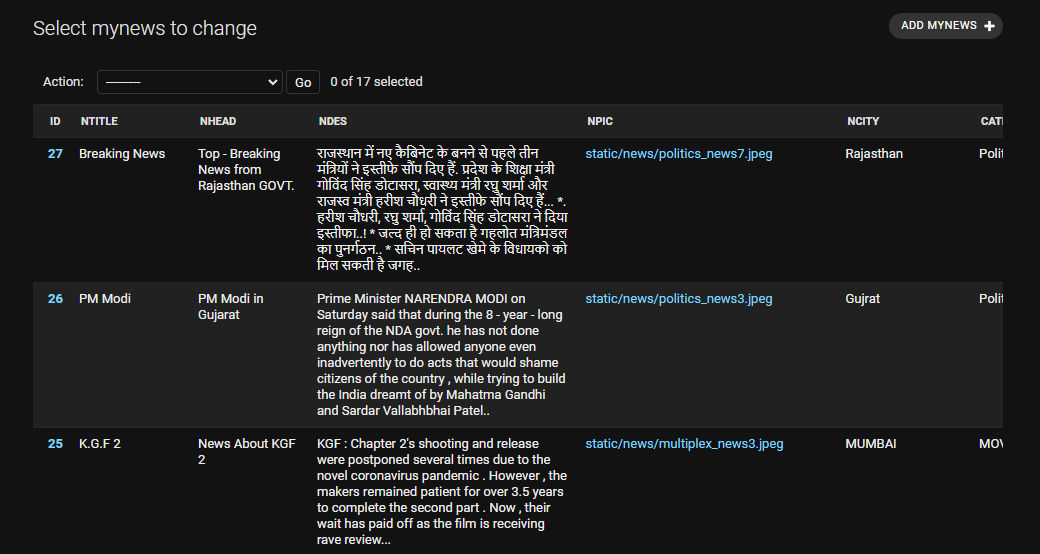
**6.2.5 IGallery Table**



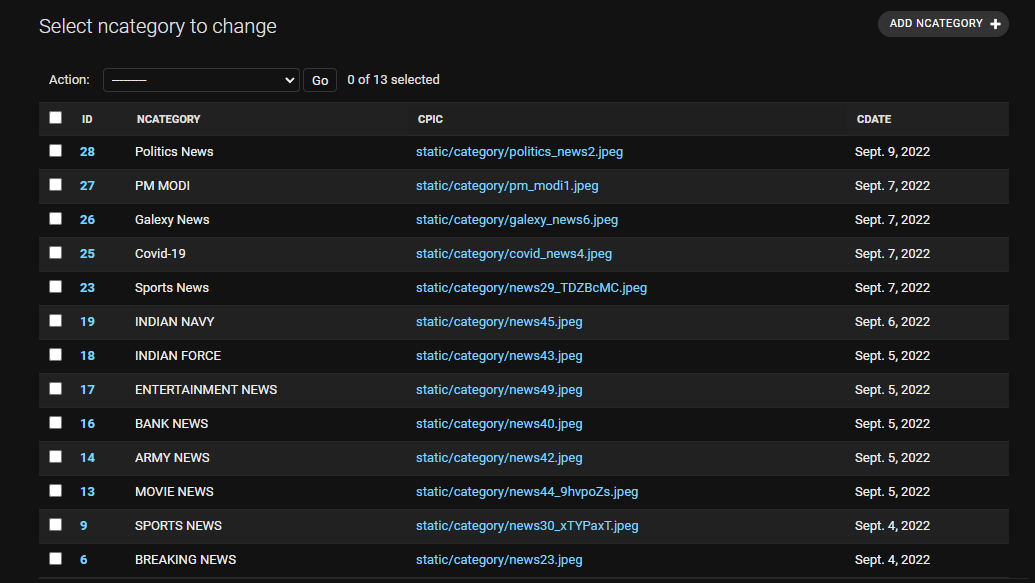
**6.2.6 MyNews Table**



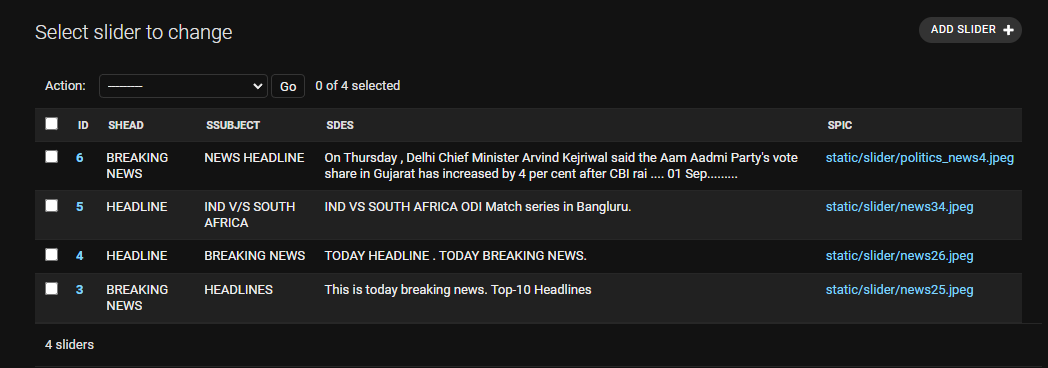




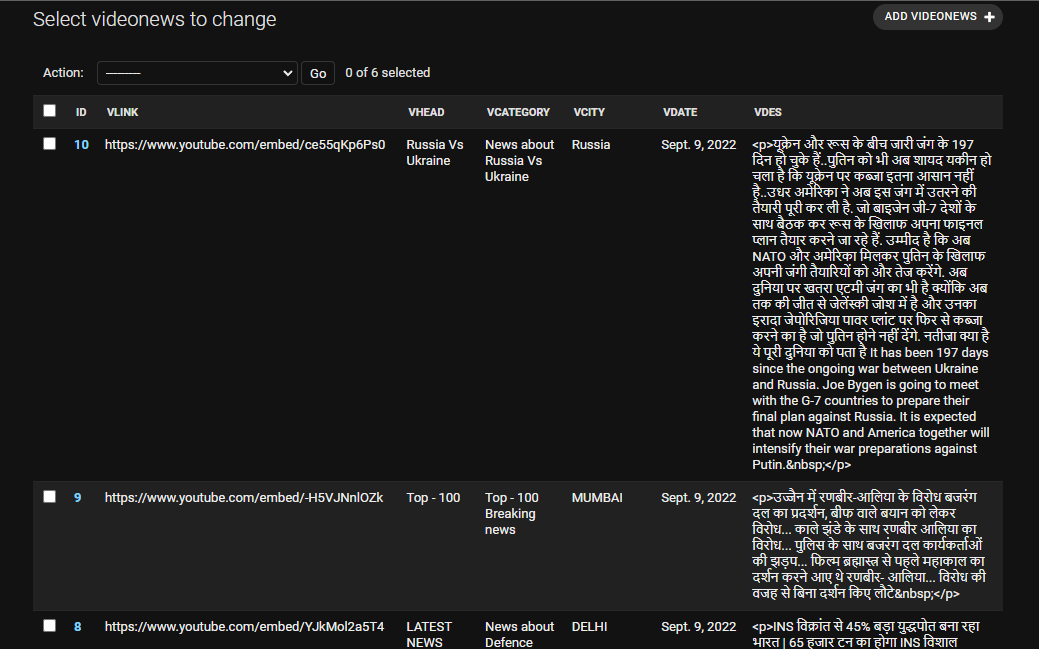
**6.2.7 NCategory Table**



**6.2.8 Slider Table**



**6.2.9 VideoNews Table**



**7. TESTING**

Testing is the integral part of any System Development Life Cycle insufficient and interested application tends to crash and result in loss of economic and manpower investment besides user’s dissatisfaction and downfall of reputation.

“Software Testing can be looked upon as one among much process, an organization performs, and that provides the last opportunity to correct any flaws in the developed system. Software Testing includes selecting test data that have more probability of giving errors.” The first step in System testing is to develop the plan that all aspect of system Complements, Correctness, Reliability and Maintainability.

Software is to be tested for the best quality assurance, an assurance that system meets the specification and requirement for its intended use and performance.

System Testing is the most useful practical process of executing the program with the implicit intention of finding errors that makes the program fail.

**Types of Testing:**

**Black Box (Functional) Testing:**

Testing against specification of system or component. Study it by examining its inputs and related outputs. Key is to devise inputs that have a higher likelihood of causing outputs that reveal the presence of defects. Use experience and knowledge of domain to identify such test cases. Failing this a systematic approach may be necessary. Equivalence partitioning is where the input to a program falls into a number of classes,

e.g positive numbers vs. negative numbers. Programs normally behave the same way for each member of a class. Partitions exist for both input and output. Partitions may be discrete or overlap. Invalid data (i.e .outside the normal partitions) is one or more partitions that should be tested.

Internal System design is not considered in this type of testing. Tests are based on requirements and functionality.

This type of test case design method focuses on the functional requirements of the software, ignoring the control structure of the program. Black box testing attempts to find errors in the following categories:

* + Incorrect or missing functions.
  + Interface errors.
  + Errors in data structures or external database access.
  + Performance errors.
  + Initialization and termination errors.

**White Box (Structural) Testing:**

Testing based on knowledge of structure of component (e.g., by looking at source code). Advantage is that structure of code can be used to find out how many test case need to be performed. Knowledge of the algorithm (examination of the code) can be used to identify the equivalence partitions. Path testing is where the tester aims to exercise every independent execution path through the component. All conditional statements tested for both true and false cases. If a unit has statements, there will be possible paths through it This demonstrates that it is much easier to test small

program units than large ones. Flow graphs are a pictorial representation of the paths of control through a program (ignoring assignments, procedure calls and I/O statements). Use flow graph to design test cases that execute each path. Static tools may be used to make this easier in programs that have a complex branching structure. Tools support. Dynamic program analysers, instrument a program with additional code. Typically, this will count how many times each statement is executed. At end print out report showing which statements have and have not been executed. Problems with flow graph derived testing:

* Data complexity could not take into account.
* We cannot test all paths in combination.
* In really only possible at unit and module testing stages because beyond that complexity is too high.

This testing is based on knowledge of the internal logic of an application’s code. Also known as a Glass Box Testing Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions.

**Unit Testing:**

Unit testing concentrates on each unit of the software as implemented in the code. This is done to check syntax and logical errors in programs. At this stage, the test focuses on each module individually, assuring that it functions properly as a unit. In our case, we used extensive white-box testing at the unit testing stage.

A developer and his team typically do the unit testing do the unit testing is done in parallel with coding; it includes testing each function and procedure.

**Incremental Integration Testing:**

Bottom-up approach for testing i.e. continuous testing of an application as new functionality is added; Application functionality and modules should be independent enough to test separately done by programmers or by testers.

**Integration Testing:**

Testing of integration modules to verify combined functionality after integration

.Modules are typically code modules, individual applications, client and server and distributed systems.

**Functional Testing:**

This type of testing ignores the internal parts and focus on the output is as per requirement or not .Black box type testing geared to functionality requirements of an application.

**System Testing:**

Entire system is tested as per the requirements. Black box type test that is based on overall requirement specifications covers all combined parts of a system.

**End-to-End Testing:**

Similar to system testing involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database using network communications, or interacting with hardware, applications, or system if appropriate.

**Regression Testing:**

Testing the application as a whole for the modification in any module or functionality. Difficult to cover all the system in regression testing so typically automation tools are used for these testing types.

**Acceptance Testing:**

Normally this type of testing is done to verify if system meets the customer specified requirements. User or customers do this testing to determine whether to accept application.

**Performance Testing:**

Term often used interchangeably with “stress” and “load” testing, To check whether system meets performance requirements, Used different performance and load tools to do this.

**Alpha Testing:**

In house virtual user environment can be created for this type of testing. Testing is done at the end of development Still minor design changes may be made as a result of such testing.

**Beta Testing:**

Testing typically done by end-users or others. This is final testing before releasing application for commercial purpose.

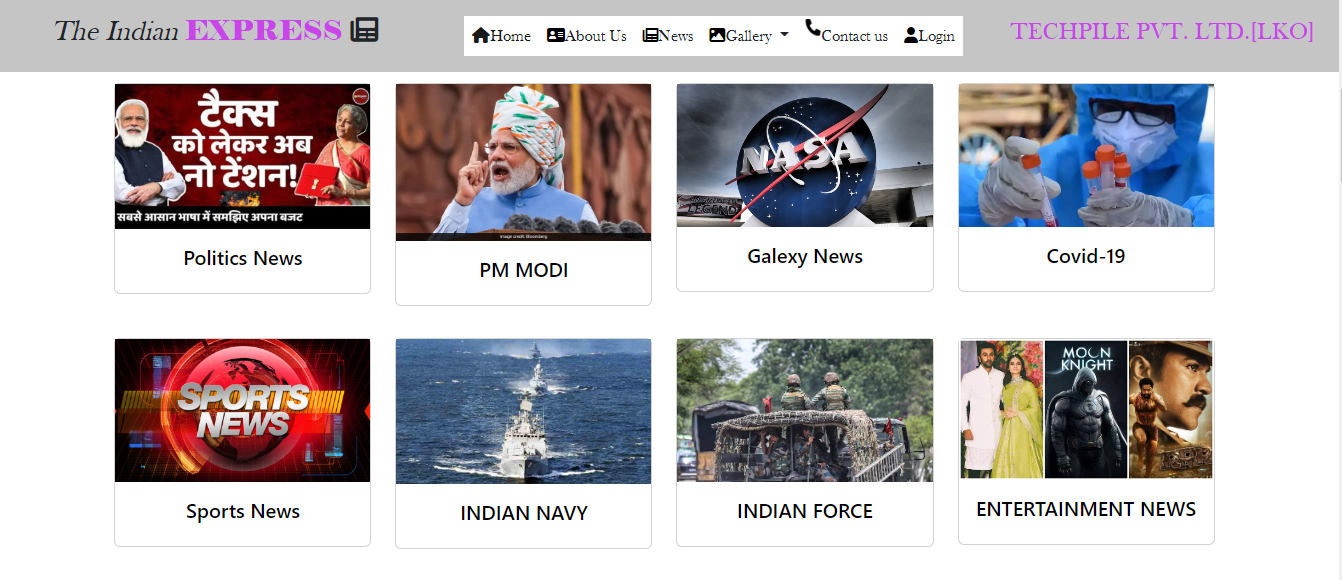
# 

**8. Input-output Forms(SCREENSHOTS AND CODING)**

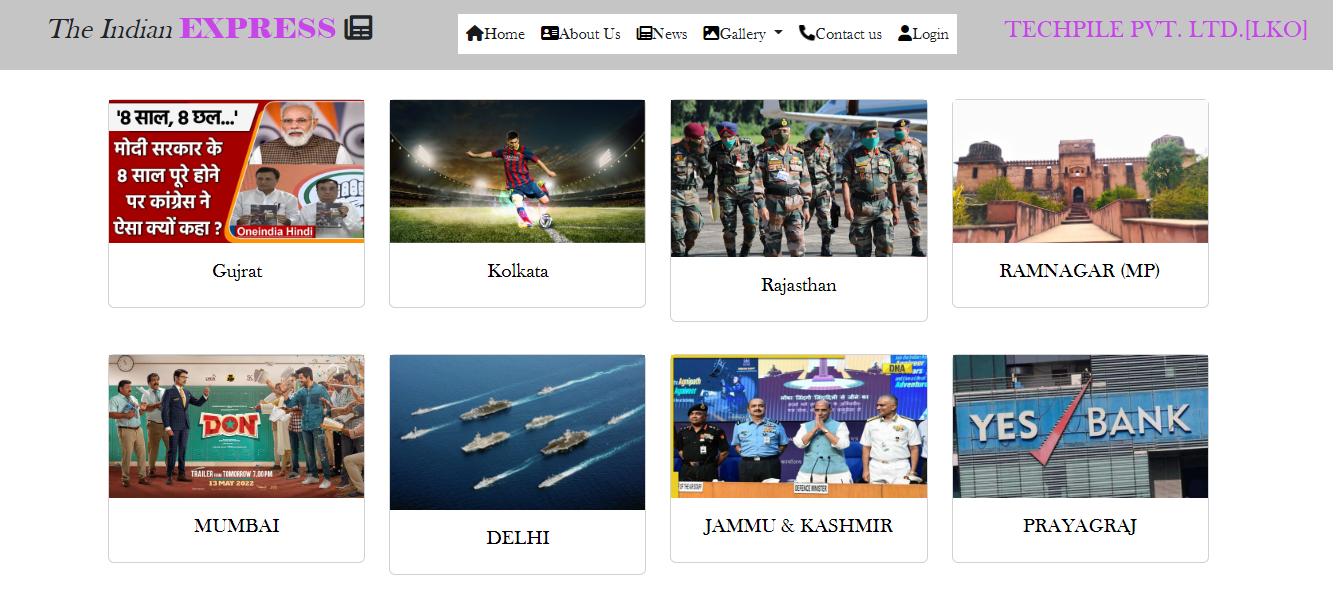
**8.1 project Screenshot**

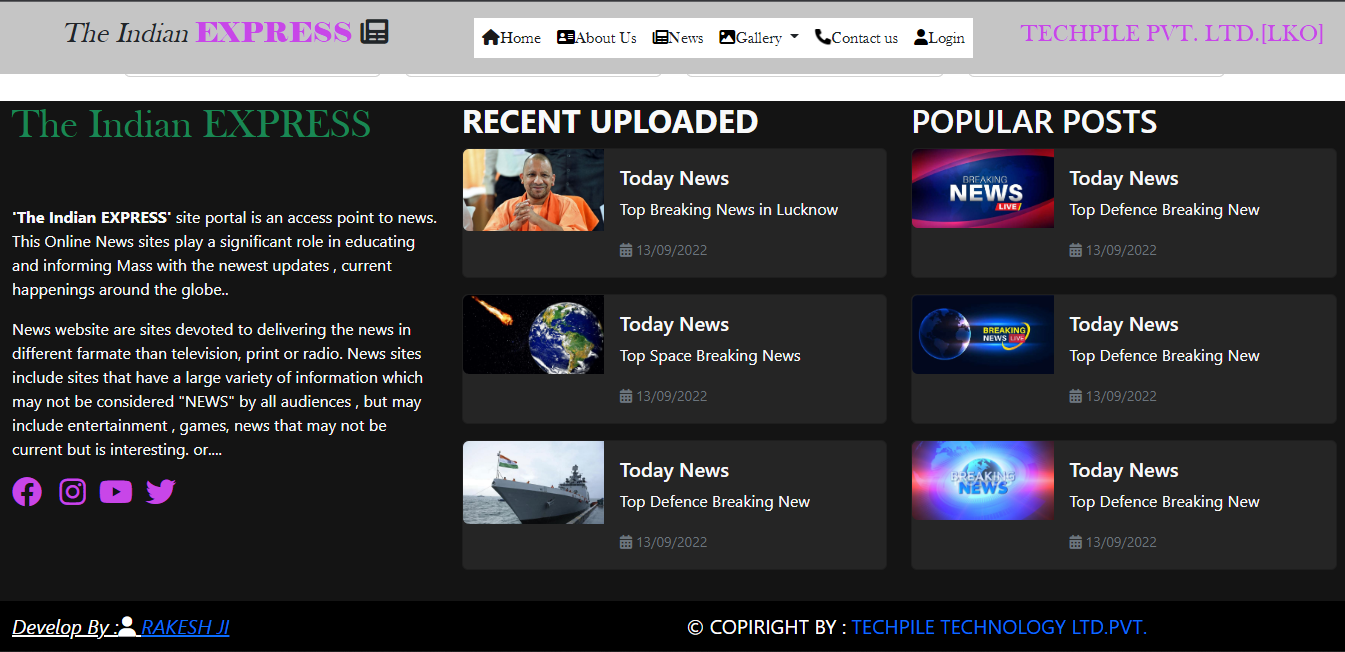
**INDEX**



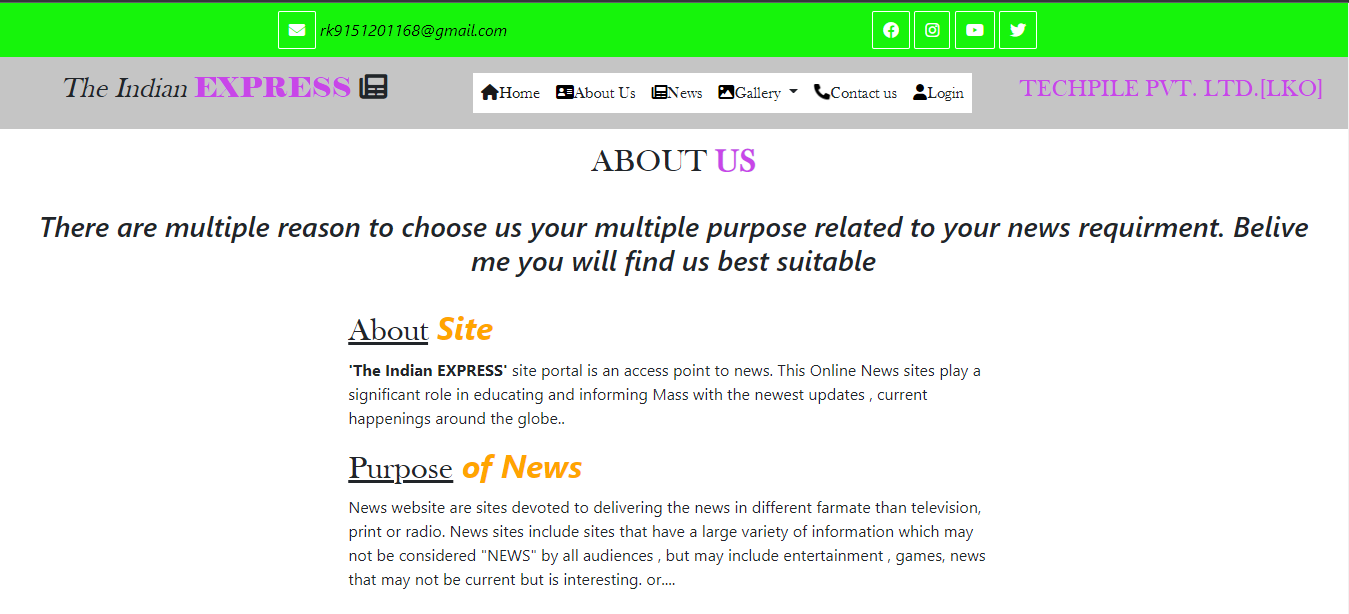


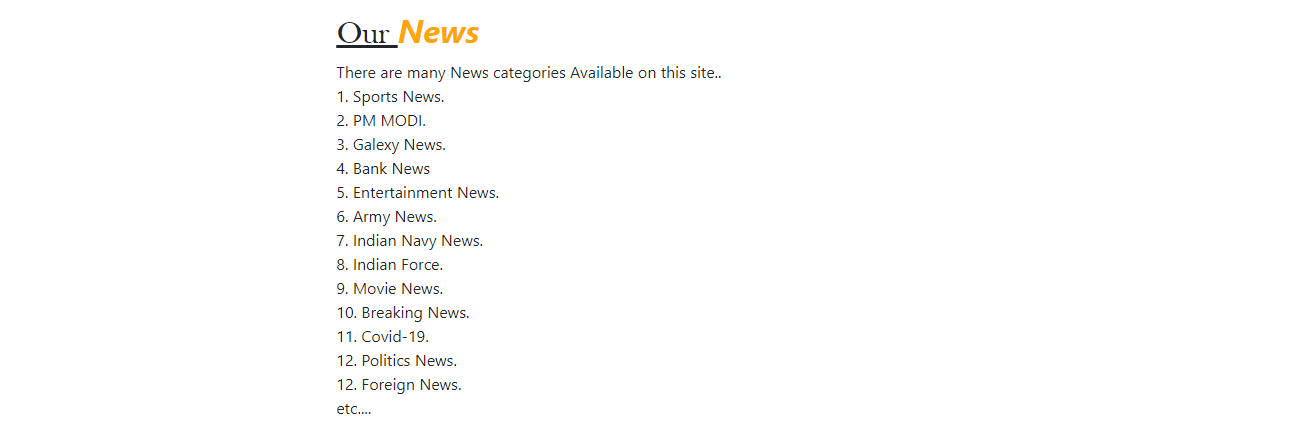




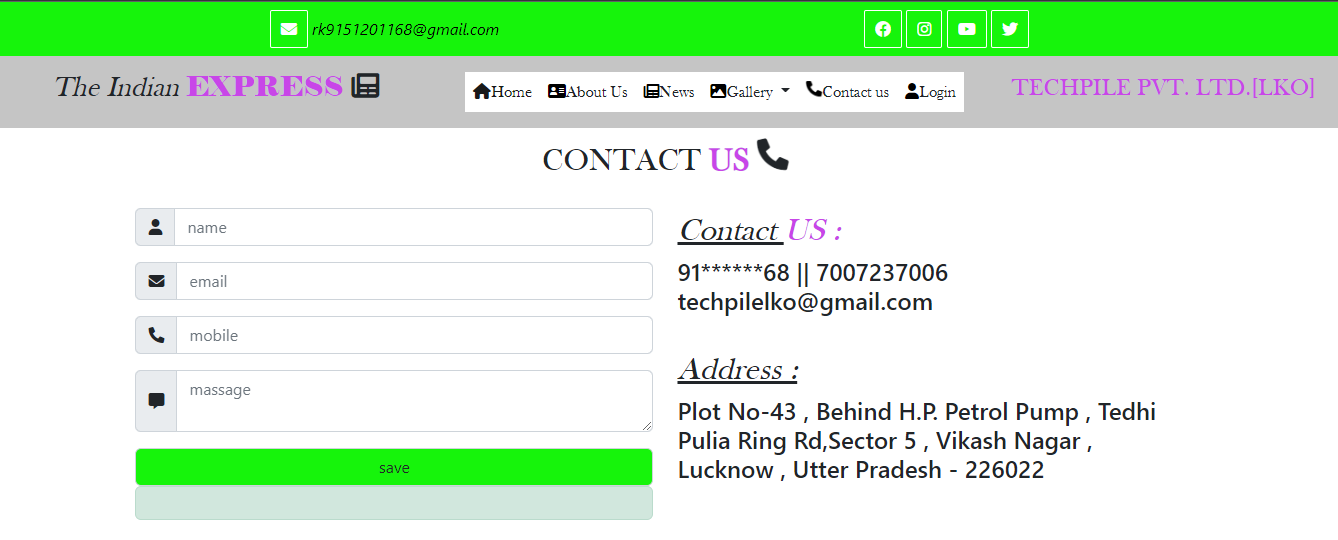


**ABOUT**





**Contact US**



**8.2 project Coding**

**coding (.html)**

**INEDX.html**

{% extends 'base.html' %}  
{% load static %}  
{% block title %}  
The Indian EXPRESS  
{% endblock %}  
  
{% block x %}  
<div class="row slider ">  
 <div class="col-sm-7 s1 p-0">  
 <!--start slider -->  
<div id="carouselExampleCaptions" class="carousel slide" data-bs-ride="true">  
 <div class="carousel-indicators">  
 <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="0" class="active" aria-current="true" aria-label="Slide 1"></button>  
 <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="1" aria-label="Slide 2"></button>  
 <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="2" aria-label="Slide 3"></button>  
 </div>  
 <div class="carousel-inner">  
 <div class="carousel-item active">  
 <img src="{% static 'images/news9.jpeg' %}" class="d-block w-100" alt="...">  
 <div class="carousel-caption d-none d-md-block">  
  
 </div>  
 </div>  
 <!--for loop -->  
 {% for a in res %}  
 <div class="carousel-item">  
 <img src="/{{a.spic}}" class="d-block w-100" alt="...">  
 <div class="carousel-caption d-none d-md-block">  
 <h3 class="text-dark">{{a.shead}}</h3>  
 <p class="text-danger">{{a.ssubject}}</p>  
 </div>  
 </div>  
 {% endfor %}  
  
 <!-- end for loop -->  
  
 </div>  
 <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide="prev">  
 <span class="carousel-control-prev-icon" aria-hidden="true"></span>  
 <span class="visually-hidden">Previous</span>  
 </button>  
 <button class="carousel-control-next" type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide="next">  
 <span class="carousel-control-next-icon" aria-hidden="true"></span>  
 <span class="visually-hidden">Next</span>  
 </button>  
</div>  
 <!-- end slider -->  
  
 </div>  
 <div class="col-sm-5 s2 " style="height:442px;overflow-x:auto;">  
 <marquee direction="up" height="470px" behavior="alternate">  
 <!-- start card -->  
  
 {% for x in ndata %}  
  
 <div class="card mb-3 w-100">  
 <div class="row g-0">  
 <div class="col-md-4">  
 <img src="/{{x.npic}}" class="img-fluid rounded-start" style="min-height:10px;" alt="...">  
 </div>  
 <div class="col-md-8">  
 <div class="card-body">  
 <h5 class="card-title">{{x.nhead}}</h5>  
 <a href="/user/details?n={{x.id}}"><input type="button" value="VIEW MORE " class="btn btn-sm bgcolor text-white" /></a>  
 <p class="card-text"><small class="text-muted"><i class="fa-solid fa-calendar"> :{{x.ndate}}</i></small></p>  
 </div>  
 </div>  
 </div>  
</div>  
  
 {% endfor %}  
  
 </marquee>  
 <!-- end card -->  
  
 </div>  
  
 </div>  
 <div class="row ncategory ">  
  
 <div class="col-sm-1"></div>  
 <div class="col-sm-10 ">  
 <div class="row">  
 {% for i in data %}  
 <div class="col-sm-3 my-3">  
 <div class="card">  
 <img src="/{{i.cpic}}" class="img-card-title"/>  
 <div class="card-body">  
 <a href="/user/viewnews?msg={{i.id}}" style="text-decoration:none;color:black;">  
 <h5 class="card-title text-center ">{{i.ncategory}}</h5>  
 </a>  
 </div>  
 </div>  
 </div>  
  
 {% endfor %}  
  
 </div>  
  
  
 </div>  
 <div class="col-sm-1"></div>  
  
  
 </div>  
<div class="fs-2 text-center font"> LATEST <b class="text-mycolor" >NEWS</b>  
 <i class="fa-solid fa-image"></i>  
 <div class="row inews" style="background:#cecece;">  
  
 {% for x in ndata %}  
 <div class="col-sm-3">  
 <div class="card">  
 <img src="/{{x.npic}}" class="img-card-title" alt="path error"/>  
 <div class="card-body">  
 <h3 class="card-title text-center">{{x.ntitle}}</h3>  
 <h5 class="card-title text-center">{{x.nhead}}</h5>  
 <a href="/user/details?n={{x.id}}" style="text-decoration:none;" > <input type="button" value="VIEW MORE " class="form-control bgcolor" /></a>  
 <p class="card-text"><small class="text-muted">Date<i class="fa-solid fa-calendar"></i>28/08/2022</small></p>  
  
 </div>  
 </div>  
 </div>  
  
 {% endfor %}  
  
  
 </div>  
  
  
  
<!-- start video news -->  
  
<div class="row py-4">  
 <div class="fs-2 text-center font"> LATEST VIDEO <b class="text-mycolor" >NEWS</b>  
 <i class="fa-solid fa-image"></i>  
</div>  
 {% for i in vdata %}  
 <div class="col-sm-6">  
  
 <!-- start card -->  
  
 <div class="card mb-3" style="max-width: 540px;">  
 <div class="row g-0">  
 <div class="col-md-4">  
  
 <iframe class="w-100" src="{{i.vlink}}" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>  
  
 </div>  
 <div class="col-md-8">  
 <div class="card-body">  
 <h5 class="card-title">{{i.vcategory}}</h5>  
 <b> {{i.vhead}} </b></br>  
 <a href="/user/vdetail?vid={{i.id}}" class="btn btn-sm bgcolor text-white"> view more</a>  
 <a href="#" class="btn btn-sm bgcolor text-white float-end"> {{i.vcity}} </a>  
 <p class="card-text"> </p>  
 <p class="card-text"><small class="text-muted"> <i class="fa-solid fa-calendar-days"></i> {{i.vdate}}</small></p>  
 </div >  
 </div>  
 </div>  
</div>  
  
 <!-- end card -->  
 </div>  
 {% endfor %}  
</div>  
  
<!-- end vnews -->  
  
  
  
 <div class="row city">  
  
 <div class="col-sm-1"></div>  
 <div class="col-sm-10">  
 <div class="row category1">  
 {% for i in city %}  
 <div class="col-sm-3 my-3">  
 <div class="card">  
 <img src="/{{i.cpic}}" class="img-card-title"/>  
 <div class="card-body">  
 <a href="/user/viewnews?cid={{i.id}}" style="text-decoration:none;color:black;">  
 <h5 class="card-title text-center ">{{i.ncity}}</h5>  
 </a>  
 </div>  
 </div>  
 </div>  
  
 {% endfor %}  
  
 </div>  
 </div>  
  
 </div>  
 <div class="col-sm-1"></div>  
  
  
 </div>  
{% endblock %}

**CONTACT US.html**

{% extends 'base.html' %}  
  
  
{% block x %}  
<div class="fs-2 text-center font"> CONTACT <b class="text-mycolor" >US</b>  
 <i class= "fa-solid fa-phone fa-bounce"></i>  
</div>  
<div class="row p-4">  
 <div class="col-sm-1"> </div>  
  
 <div class="col-sm-5 frm">  
 {% if msg %}  
 <div class="alert alert-success">  
 YOUR DATA SAVED SUCCESSFULLY.  
 </div>  
 {% endif %}  
 <form method="post">  
 {% csrf\_token %}  
 <div class="input-group text-light mb-3">  
 <i class="input-group-text"><i class="fa-solid fa-user"></i></i>  
 <input name="name" type="text" required class="form-control" placeholder="name"/>  
  
 </div>  
  
 <div class="input-group bg-mycolor text-light mb-3">  
 <i class="input-group-text"><i class="fa-solid fa-envelope"></i></i>  
 <input name="email" type="text" required class="form-control" placeholder="email"/>  
  
 </div>  
  
 <div class="input-group bg-mycolor text-light mb-3">  
 <i class="input-group-text"><i class="fa-solid fa-phone"></i></i>  
 <input name="mob" type="text" required class="form-control" placeholder="mobile"/>  
  
 </div>  
  
 <div class="input-group bg-mycolor text-light mb-3">  
 <i class="input-group-text"><i class="fa-solid fa-message"></i></i>  
 <textarea name="msg" required class="form-control" placeholder="massage"></textarea>  
  
 </div>  
 <input type="submit" value="save" class="form-control bgcolor"/>  
  
 </form>  
 <div class="alert alert-success">  
 <!--  
 Name Is :{{Name}}</br>  
 Email Is :{{Email}}</br>  
 Contact Is :{{Mobile}}</br>  
 Msg Is :{{Message}}</br>  
 -->  
 </div>  
 </div>  
  
 <div class="col-sm-5">  
  
 <h2> <i class="font"> <u> Contact </u> </i> <i class="font text-mycolor"> US : </i></h2> <h4> 91\*\*\*\*\*\*68 || 7007237006  
  
 techpilelko@gmail.com  
  
 </h4></br>  
 <h2><i class="font"><u>Address : </u></i></h2>  
 <h4>Plot No-43 , Behind H.P. Petrol Pump , Tedhi Pulia  
 Ring Rd,Sector 5 , Vikash Nagar , </br>  
 Lucknow , Utter Pradesh - 226022  
 </h4>  
  
  
 </div>

**ABOUT US.html**

{% extends 'base.html' %}  
  
  
{% block x %}  
<div class="fs-2 text-center font "> ABOUT <b class="text-mycolor" >US</b>  
  
</div>  
<h3 class="text-center p-4"><i> There are multiple reason to choose us your multiple purpose  
 related to your news requirment. Belive me you will find us best suitable </i></h3>  
<div class="row">  
 <div class="col-sm-3"></div>  
 <div class="col-sm-6">  
 <h2><u class="font">About</u> <b><i style="color:orange;"> Site</i></b> </h2>  
  
 <p> <B>'The Indian EXPRESS' </B> site portal is an access point to news.  
 This Online News sites play a significant role in educating and  
 informing Mass with the newest updates , current happenings around the globe..</p>  
  
 <h2><u class="font">Purpose</u> <b><i style="color:orange;"> of News</i></b> </h2>  
 <p> News website are sites devoted to delivering the news in different farmate than television, print or radio.  
 News sites include sites that have a large variety of information  
 which may not be considered "NEWS" by all audiences , but may include entertainment , games, news  
 that may not be current but is interesting. or....  
 </p>  
 <h2><u class="font">Our </u><b><i style="color:orange;"> News</i></b></h2>  
 <p>  
 There are many News categories Available on this site..</br>  
 1. Sports News.</br>  
 2. PM MODI.</br>  
 3. Galexy News.</br>  
 4. Bank News</br>  
 5. Entertainment News.</br>  
 6. Army News.</br>  
 7. Indian Navy News.</br>  
 8. Indian Force.</br>  
 9. Movie News.</br>  
 10. Breaking News.</br>  
 11. Covid-19. </br>  
 12. Politics News. </br>  
 12. Foreign News. </br>  
 etc....  
  
  
  
 </p>  
  
  
  
  
 </div>  
 <div class="col-sm-3"></div>  
</div>  
  
{% endblock %}

## 9. FUTURE SCOPE

*Following modification or upgrades can be done in system.*

* 1. More than one company can be integrated through this software.
  2. Web services can be used to know exact donation status of packets.
  3. Client can check their donation delivery status online.

**10. CONCLUSION**

At the last the Conclusion of project is to develop a web-application which the help programmer to get help from the site, so that they can develop their project and application Different Technologies and make a group of programmer. A **Group** is a social unit of any size that shares common values, ideas and code and queries. The portal doesn’t have to be expensive. It supports multiple programmer goals.

**Samachar India** is not only a web portal; it is a live product of board of technical education. In future we will add more and more features on it.