PROJECT DOCUMENTATION

Natural Disasters Intensity Analysis and Classification Using Artificial Intelligence.

1.INTRODUCTION

1.1 PROJECT OVERVIEW

Natural Disasters are catastrophic events with atmospheric and historic origins (hurricanes, floods, tsunamis, earthquakes). That can cause fatalities, property damage and social environment disruption.

Natural disasters are the results of a hazard overwhelming highly vulnerable community,often resulting in mortality and morbidity.Over the past decade,over 300 natural disasters occur yearly around the world affecting millions and cost billions.The disaster cycle is a framework used to base a coordinated plan to respond,recover,prevent,and prepare for a disaster.Access to clean water,proper sanitation,food/nutrition,shelter,and the threat of communicable diseases are concerns that have potential to be detrimental to the management of a natural disaster,slowing the recovery process.

Natural disasters not only disturb the human ecological system but also destroy the properties and critical infrastructures of human societies and even lead to permanent change in the ecosystem. Disaster can be caused by naturally occurring events such as earthquakes, cyclones, floods, and wildfires. Many deep learning techniques have been applied by various researchers to detect and classify natural disasters to overcome losses in ecosystems, but detection of natural disasters still faces issues due to the complex and imbalanced structures of images. To tackle this problem, we developed a multilayered deep convolutional neural network model that classifies the natural disaster and tells the intensity of disaster of natural The model uses an integrated webcam to capture the video frame and the video frame is compared with the Pre-trained model and the type of disaster is identified and showcased on the OpenCV window.

1.2 PURPOSE

Basically the main objective of natural disaster management is to reduce the damage. However, there are several objectives that are integrated with it. Those are,

1. Identifying the hazard and its cause.

2. Reducing vulnerability and potential losses of hazard.

3. Assessing, reviewing and controlling the risk.

4. Applying efficient, effective, sustainable relief (food, shelter and money), medical and other facilities in disaster affected people thus they can survive.

5. Reducing the damage, death, sufferings and destruction of any natural and human induced disaster.

6. Giving protection to victims.

7. Increasing the strength among people to survive against disasters.

8. Building up capacity in every sector like- individual, social, economic, environmental, regional, national and international.

9. Ensuring the availability of local emergency equipment and transportation.

10.Promote the culture of disaster risk prevention and mitigation at all levels.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

Natural disasters not only disturb the human ecological system but also destroy the properties and critical infrastructures of human societies and even lead to permanent change in the ecosystem. Disaster can be caused by naturally occurring events such as earthquakes, cyclones, floods, and wildfires. Many deep learning techniques have been applied by various researchers to detect and classify natural disasters to overcome losses in ecosystems, but detection of natural disasters still faces issues due to the complex and imbalanced structures of images.

2.2 REFERENCES

[1. Amit S.N.K.B., Aoki Y. Disaster detection from aerial imagery with convolutional neural network; Proceedings of the2017 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC); Surabaya,Indonesia.

2. Padmawar P.M., Shinde A.S., Sayyed T.Z., Shinde S.K., Moholkar K. Disaster Prediction System using Convolution NeuralNetwork; Proceedings of the 2019 International Conference on Communication and Electronics Systems (ICCES);Coimbatore, India.

3. Nguyen D.T., Ofli F., Imran M., Mitra P. Damage assessment from social media imagery data during disasters;Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining; Sydney,NSW, Australia.

4. D. Han, L. Chan, and N. Zhu, ``Flood forecasting using support vector machines,".

5. Tonini M., D'Andrea M., Biondi G., Degli Esposti S., Trucchia A., Fiorucci P. A Machine Learning-Based Approach forWildfire Susceptibility Mapping. The Case Study of the Liguria Region in Italy

6. X. H. Le, H. V. Ho, G. Lee, and S. Jung, ``Application of long short-term memory (LSTM) neural network for flood forecasting"

7. M. F. Piñeros, E. A. Ritchie, and J. S. Tyo, "Estimating tropical cyclone intensity from infrared image data,"

2.3 PROBLEM STATEMENT DEFINITION

The purpose of the problem statement is to identify the issue that is a concern and focus it in a way that allows it to be studied in a systematic way. It defines the problem and proposes a way to research a solution, or demonstrates why further information is needed in order for a solution to become possible.

Problem Statement is inclusive of below answers:

- Who does it affect/does not affect?
- What does it affect/does not affect?
- How does it affect/does not affect?
- When is it a problem/is not a problem.
- Where is it a problem/is not a problem.

Natural disasters not only disturb the human ecological system but also destroy the properties and critical infrastructures of human societies and even lead to permanent change in the ecosystem. Disaster can be caused by naturally occurring events such as earthquakes, cyclones, floods, and wildfires. Many deep learning techniques have been applied by various researchers to detect and classify natural disasters to overcome losses in ecosystems, but detection of natural disasters still faces issues due to the complex and imbalanced structures of images.

The natural disasters disturbs the ecosystem,both humans and animals.(WHO)

In a disaster, **you face the danger of death or physical injury**. You may also lose your home, possessions, and community. Such stressors place you at risk for emotional and physical health problems. Stress reactions after a disaster look very much like the common reactions seen after any type of trauma.(WHAT)
Wildlife can be killed by the force of the disaster or impacted indirectly through changes in habitat and food availability. Endangered species are especially vulnerable when habitat is destroyed. Water quality is impacted when sewage treatment facilities flood or debris enters reservoirs and waterways.(HOW)

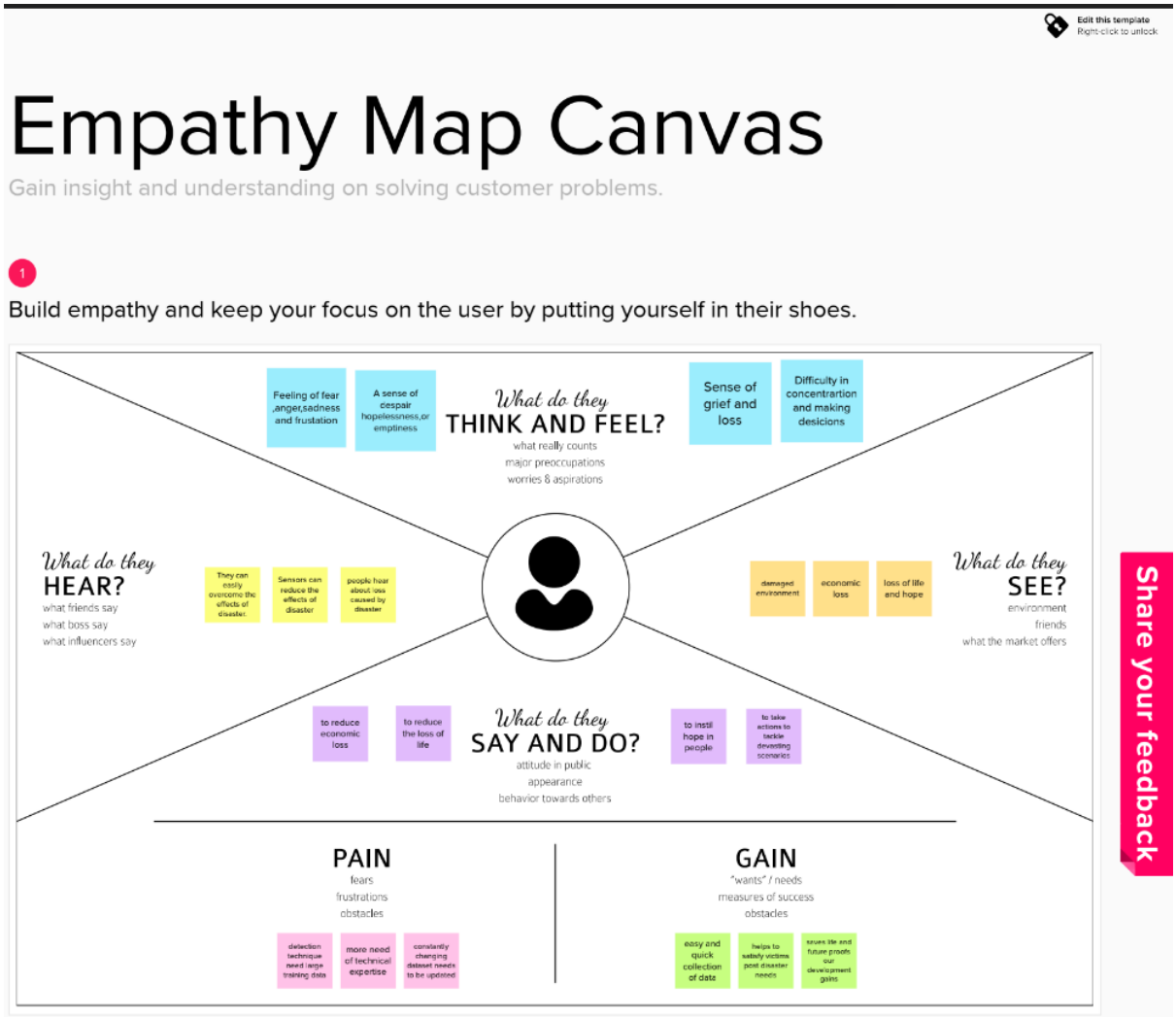
Natural disasters generally constitute an emergency since **they require immediate intervention due to their high impact on human health and safety**; they affect the normal functioning of working infrastructure, interrupting normal day activities and representing a risk for residents and workers in affected areas.(WHEN)

An area is defined as a hazard prone area **if the mortality risk is higher than a certain threshold**. A cyclone is defined as a wind storm with a maximum speed of more than 64 knots per hour. The definition includes typhoons and hurricanes.(WHERE)

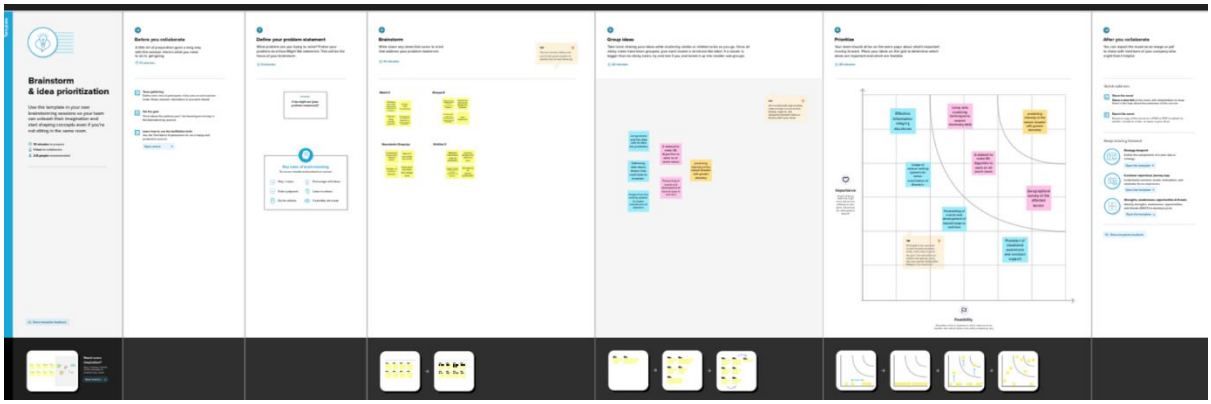
Problem Statement (PS)	**I am (Customer)**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	Human(People)	Avoid the natural disaster	Due to natural disasters, there are droughts, economic crises, capital destruction etc.	Natural disasters are increasing because of population growth,Urbanisation(a lot of people in small places), alteration of the natural environment(man-made islands)	Natural disasters affect human life and destroy natural resources.

3.IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION AND BRAINSTROMING



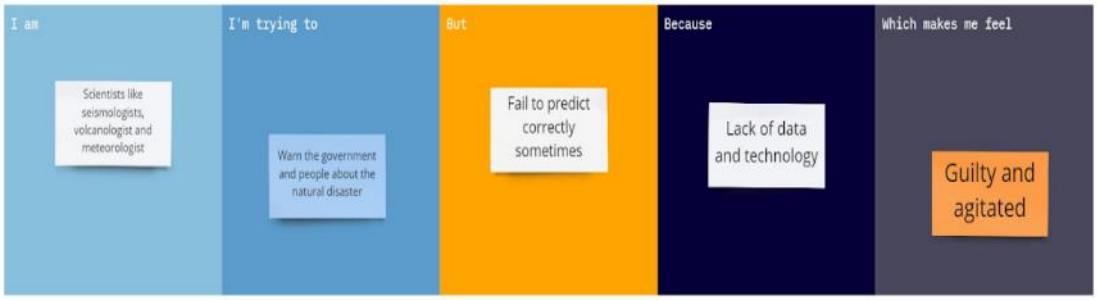
3.3 PROPOSED SOLUTION

Project team shall fill the following information in the proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Disaster can be caused by naturally occurring events such as earthquakes, cyclones, floods, and wildfires. Many deep learning techniques have been applied by various researchers to detect and classify natural disasters to overcome losses in ecosystems, but detection of natural disasters still faces issues due to the complex and imbalanced structures of images
2.	Idea / Solution description	The proposed model shows better accuracy as compared to the recently developed techniques. The reason for this is that the proposed technique works in two parts: one for natural disaster occurrence detection and the second one for natural disaster classifications.
3.	Novelty / Uniqueness	We proposed multilayered deep convolutional neural network for detection and intensity classification of natural disasters.
4.	Social Impact / Customer Satisfaction	Intensity of the disaster can be measured, which is useful in taking further precautions to avoid damage.
5.	Scalability of the Solution	Natural disasters not only disturb the human ecological system and destroy the properties but also causes death of people. By knowing the Intensity. we can make people vacate to safe places.

3.4 PROBLEM SOLUTION FIT

PROBLEM STATEMENT 1:



PROBLEM STATEMENT 2:

4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registering via Google Accounts and Registering via Product's own user management system
FR-2	User Authentication	Verification through OTP and Verification through Email Link
FR-3	Designation of Region	Ease of selection of necessary areas to be monitored and Versatile and Flexible operations on designated areas

FR-4	Analysis of Required Phenomenon	Simple and easy analysis on the specific phenomenon to be observed
FR-5	Organizing Unstructured data	Processing of raw and clustered data into clear and refined data which is useful for analysis and prediction tasks
FR-6	Algorithm selection	The freedom to choose from several classes of algorithm to be used in the process and Customization of algorithm to suit the needs of a specific purpose
FR-7	Prediction and analysis of data	Accurate results of the analysis provided by the process and Advanced visualization techniques to help visualize the processed data for effective observation
FR-8	Report generation	Restructuring of obtained results into clear and detailed report for future studies

4.2 NON-FUNCTIONAL REQUIREMENT

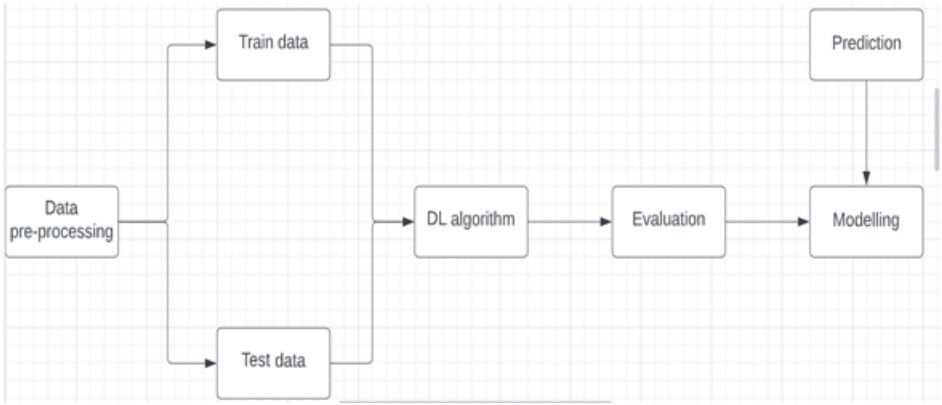
Following are the non-functional requirements of the proposed solution.

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	User friendly and classify the disaster easily
NFR-2	**Security**	It provides a distinctand secure encryption layer to the system interface for additional security standards.
NFR-3	**Reliability**	The product is robust and is capable of execution of processes even in the most difficult and unpredictable environments.

NFR-4	**Performance**	The product boasts a high precision and efficient working capacity which helps in escalating its performance to the highest degree
NFR-5	**Availability**	Despite the complexity and degree of difficulty in its operation, the product is equipped with all-round maintenance and readily available technical services which provides the necessary support any individual requires in their duties
NFR-6	**Scalability**	The product also possess enough room for the improvement of its specifications to upgrade its capabilities according to the needs of the user and their organization

5.PROJECT DESIGN

5.1 DATA FLOW GRAPHS



5.2 SOLUTION AND TECHNICAL ARCHITECTURE

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to: ′ Find the best tech solution to solve existing business problems. ′ Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders. ′ Define features, development phases, and solution requirements. ′ Provide specifications according to which the solution is defined, managed, and delivered.

Solution Architecture Diagram:

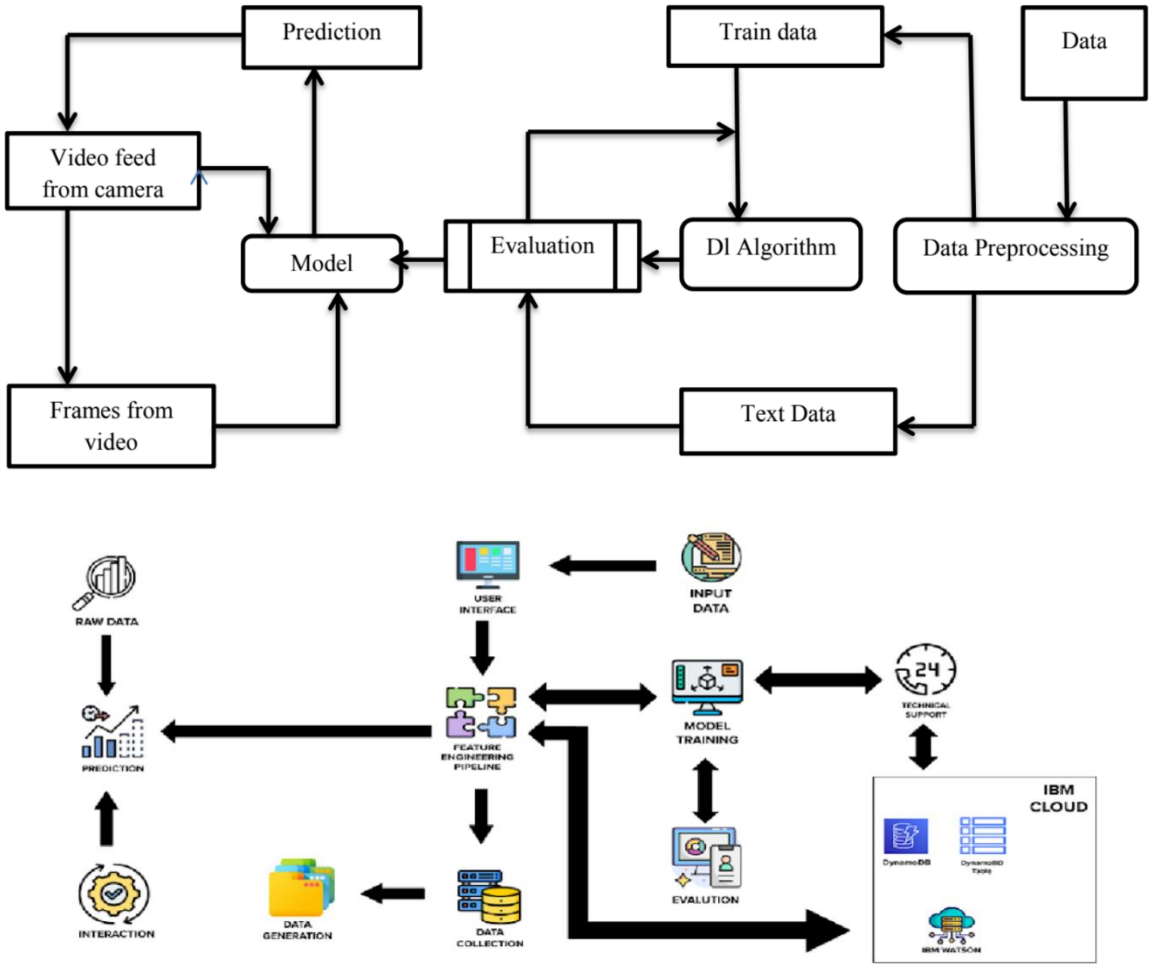


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	User interacts with application for the prediction of Any Natural disaster which will happen in future minutes.	HTML, CSS, JavaScript, Django, Python
2.	Feature Engineering Pipeline	Algorithms can't make sense of raw data. We have to select, transform, combine, and otherwise prepare our data so the algorithm can find useful patterns.	Image processing, pattern extraction, etc.

3.	Model Training kit	It learns patterns from the data. Then they use these patterns to perform particular tasks	Multiclass Classification Model, Regression Model, etc.
4.	Prediction unit	This function is used to predict outcomes from the new trained data to perform new tasks and solve new problems.	Decision trees, Regression, Neural networks.
5.	Evaluation system	It monitors that how Algorithm performs on data as well as during training.	Chi-Square, Confusion Matrix, etc
	Interactive services	To interact with our model and give it problems to solve. Usually this takes the form of an API, a user interface, or a command-line interface.	Application programming interface, etc.
7.	Data collection unit	Data is only useful if it's accessible, so it needs to be stored ideally in a consistent structure and conveniently in one place.	IBM Cloud, SQL Server.
8.	Data generation system	Every machine learning application lives off data. That data has to come from somewhere. Usually, it's generated by one of your core business functions.	Synthetic data generation.
9.	Database management system	An organized collection of data stored in database, so that it can be easily accessed and managed.	MySQL, DynamoDB etc
10.	IBM Cloud services	Processed data stored in cloud service which can be access by the admin anywhere over the internet	BM Cloud etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	An open source framework is a template for software development that is designed by a social network of software developers. These frameworks are free for public use and provide the foundation for building a software application.	Keras, pensor flow.
2.	Authentication	This keeps our models secure and makes sure only those who have permission can use them	Encryption and Decryption (OTP).
3.	Application interface	User uses mobile application and web application to interact with model	Android and Web Development (PhoneGap, ReactNative, and NativeScript).
4.	Availability (both Online and Offline work)	Its include both online and offline work. As good internet connection is need for online work to explore the software perfectly. Offline work includes the saved data to explore for later time.	Caching, backend server
5.	Regular Updates	The truly excellent software product needs a continuous process of improvements and updates. Maintain your server and make sure that your content is always up-to-date. Regularly update an app and enrich it with new features.	• Waterfall Approach • Incremental Approach • Spiral Approach
6.	Personalization	Software has features like flexible fonts, backgrounds, settings, colour themes, etc. which make a software interface looks good and functional.	• HubSpot • Proof

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Installation	USN-1	As a user, I can install this where the disaster occur	I can do it by myself	High	Sprint-1
customer	Designation of Region	USN-2	I can select the region of interest to be monitored and analyzed	I can choose certain specific places without error	High	Sprint-1
customer	Analysis of Required Phenomenon	USN-3	I am able to monitor certain factors that influence the actions of the phenomenon	I can monitor most of the factors involved in the action	High	Sprint-2

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
customer	Accumulation of required Data	USN-4	I am able to gather data regarding past events and a detailed report on past analysis	I can allow the storage of data of past events for certain extent	Medium	Sprint-2
customer	Algorithm selection	USN-5	I am able to choose the required algorithm for a specific analysis	I can choose various options for the algorithm to be used	High	Sprint-2
customer	Prediction and analysis of data		I am able to easily predict and visualize the data	I can use easy use prediction and visualization techniques	High	Sprint - 3
Customer (Web user)	Report generation		I am able to generate a clear and detailed report on the analysis	I can generate Report fast and it would be efficient and not complex	Medium	Sprint - 4

6.PROJECT PLANNING AND SCHEDULING

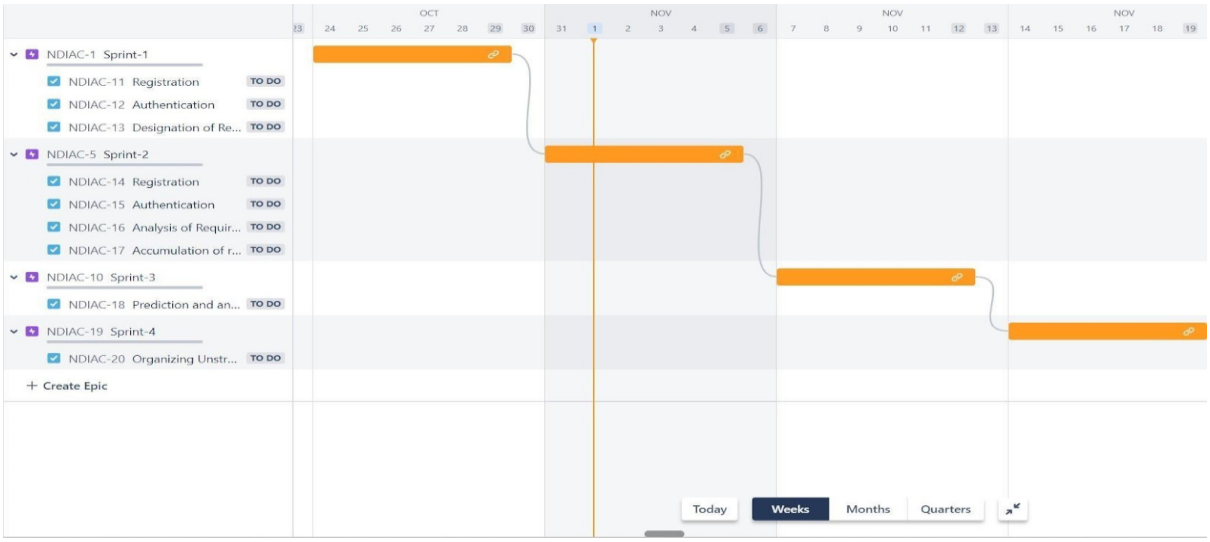
6.1 SPRINT PLANNING AND ESTIMATION

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.	2	High	Malini .V
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Khavya .N
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	Kirthika .V
Sprint-2		USN-4	As a user, I can register for the application through Gmail	2	Medium	Neerukattu . sivapriya
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Malini .V
Sprint-1	Dashboard	USN-6	As a user, I can access the services and information provided in the dashboard	2	High	Khavya .N
Sprint-1	login	USN-7	As a user, I can log into the web application and access the dashboard	2	High	Kirthika .V
Sprint-4	Helpdesk	USN-8	As a user, I can get the guidance from the customer care	1	High	Neerukattu . sivapriya
Sprint-3	Management	USN-9	As an administrator, I can collect new datasets and keep the model trained	2	High	Malini .V
Sprint-3		USN-10	As an administrator, I can update other features of the application	2	Medium	Khavya .N
Sprint-3		USN-11	As an administrator, I can maintain the information about the user	2	Medium	Kirthika .V
Sprint-4		USN-12	As an administrator, I can maintain thirdparty services	1	Low	Neerukattu . sivapriya

6.2 SPRINT DELIVERY SCHEDULE

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	8	6 Days	26 Oct 2022	31 Oct 2022	8	29 Oct 2022
Sprint-2	4	6 Days	1 Oct 2022	05 Nov 2022	4	05 Nov 2022
Sprint-3	6	6 Days	6 Nov 2022	10 Nov 2022	6	12 Nov 2022
Sprint-4	2	6 Days	10 Nov 2022	13 Nov 2022	2	19 Nov 2022

6.3 REPORTS FROM JIRA



7.CODING AND SOLUTIONING

7.1 FEATURE 1

A convolutional neural network is a class of artificial neural networks. It is a Deep Learning algorithm that can take in an input image, assign importance to various objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms.
The advantage of CNNs is to provide an efficient dense network which performs the prediction or identification efficiently.

7.2 FEATURE 2

We developed a multilayered deep convolutional neural network model that classifies the natural disaster and tells the intensity of disaster of natural The model uses an integrated webcam to capture the video frame and the video frame is compared with the Pre-trained model and the type of disaster is identified and showcased on the OpenCV window. A multilayer neural network with appropriate weights has been shown **to be able to approximate any input-output function making it an attractive tool for modeling and forecasting**.

8.TESTING

8.1 TEST CASES

8.2 USER ACCEPTANCE TESTING

This document serves as a quick reference for the Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy project's test coverage and open issues as of the project's release for user acceptance testing.

Defect Analysis:-

This shows how many bugs were fixed or closed at each severity level and how they were fixed.

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	4	5	2	3	14
Duplicate	1	0	3	1	5
External	2	3	0	1	6
Fixed	9	2	4	15	30
Not Reproduced	0	0	1	0	1
Skipped	1	0	1	1	3
Won't Fix	0	5	2	1	8
Totals	17	14	13	22	64

Test-Case Analysis

This report shows the number of test cases that have passed, failed, and untested.

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	9	0	0	9
Client Application	40	0	0	40
Security	3	0	0	3
Out-source Shipping	3	0	0	3
Exception Reporting	8	0	0	8
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9.RESULTS

9.1 PERFORMANCE METRICS

S.No.	Parameter	Values(Percentage)
1.	Model Summary	-96%
2.	Accuracy	Training Accuracy - 96.5% Validation Accuracy -92.3%
3.	Confidence Score (Only Yolo Projects)	Class Detected - Nil Confidence Score - Nil

Our Project marks the successive performance by implementing in order to be cost effective and more reliable to use and to predict the future from the natural disaster that we are ahead of. The successive way includes the objectives, activities and the approaches for the project. It mainly includes the trained dataset which gives an excessive measure of success which helps to overcome the future from this natural disaster.

10. ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

- The use of AI to forecast natural disasters would save millions of lives. Furthermore, the information evaluated by AI-powered systems can aid in understanding the scale and patterns of natural catastrophes such as floods, earthquakes, and tsunamis, which would aid in improved infrastructure development in disaster-prone areas.

- Disaster management plays an integral role in keeping communities safe. It involves coordinating the resources, such as pollution control systems, and responsibilities, such as following best practice policies, needed to prevent, prepare for, respond to, and recover from emergency

DISADVANTAGES:

- A forest fire is a natural disaster that cannot be forecasted.

- Sometimes the prediction may fail and result in huge loss.

11. CONCLUSION

Natural disasters inflict severe damage on almost the entire spectrum of social and natural habitats, ranging from housing and shelter, water, food, health, sanitation, and waste management to information and communication networks, supply of power and energy, and transportation infrastructure. The major challenges faced in all disasters include pre-disaster early warning

infrastructure; the supply of food and clean drinking water; health and sanitation; information and communication; power and energy for lighting and cooking; waste collection and disposal, including rapid disposal of dead bodies of humans and animals; disaster-proof housing and shelter; emergency and post-disaster shelters; rescue and relief operations; and transport infrastructure. Though it is not possible to prevent most of the disasters, still their effects can be alleviated or mitigated in magnitude by anticipated preparedness. Advanced disaster management technology could provide a critical support system for disaster management authorities at times of disaster-related crises. Such a technology also provides important inputs for any disaster management plan of action in modern times. Communities and individuals have to be educated on pre-disaster planning and preparedness. Awareness must be created amongst masses, for which first-aid training at grass roots level is essential. There should be a National Disaster Plan that defines the tasks of the communities and local health personnel.

For the evaluation of the model ROC and 30% landslide point's residual was used. The results showed that the accuracy of the model was estimated by ANFIS in the study area of 77.48% (good accuracy); this data-mining method depends on the number of pairs of training, experimental and fuzzy data used in the research, and, in particular, by increasing the number of fuzzy rules of the process, more accurate simulation can be provided. So far, different methods have been proposed for landslide susceptibility zonation. The accuracy or the error of each of these methods, as well as the use and comparison of each of these methods, requires knowledge of the foundations on which the methods are based. So far, different approaches to landslide susceptibility zonation have been proposed, but what is certain is that all these methods can provide accurate results with minimal data and costs and at very low levels. Combining these models with GIS and RS systems not only increases the accuracy of dealing with complex issues and uncertainties, but also leads to the emergence and development of new theories and methods in a variety of issues.

12.FUTURE SCOPE

The term "Natural Disaster" encompasses the complete realm of disaster-related activities. Traditionally people tend to think of disaster management only in terms of the post-disaster actions taken by relief and reconstruction officials; yet disaster management covers a much broader scope, and many modern disaster managers may find themselves far more involved in pre-disaster activities than in post-disaster response. Those are:

1. The refugee field of disaster management is highly specialized and requires not only many development skills but also a broader awareness of political, legal, and humanitarian issues.
2. DM aims and objectives, elements, Natural/man-made Disasters, Victims, Relief Systems.
3. Phases of Disaster Response/Relief Operations, Government's Role.

To Safeguard and make available vital materials, supplies and equipment to ensure the safety and recovery of records from predictable disasters.To reduce the risk of disasters caused by human error, deliberate destruction, and building or equipment failures. Be better prepared to recover from a major natural catastrophe.

In this project we help to build preparedness for threats and hazards by providing a low-risk, cost-effective environment to: **Test and validate plans, policies, procedures and capabilities**. Identify resource requirements, capability gaps, strengths, areas for improvement, and potential best practices. Disaster management aims to reduce, or avoid, the potential losses from hazards, assure prompt and appropriate assistance to victims of disaster, and achieve rapid and effective recovery. Disaster Risk Management includes the sum total of all activities, programmes and measures which can be taken up before, during and after a disaster with the purpose to avoid a disaster, reduce its impact or recover from its losses.

13.APPENDIX

SOURCE CODE

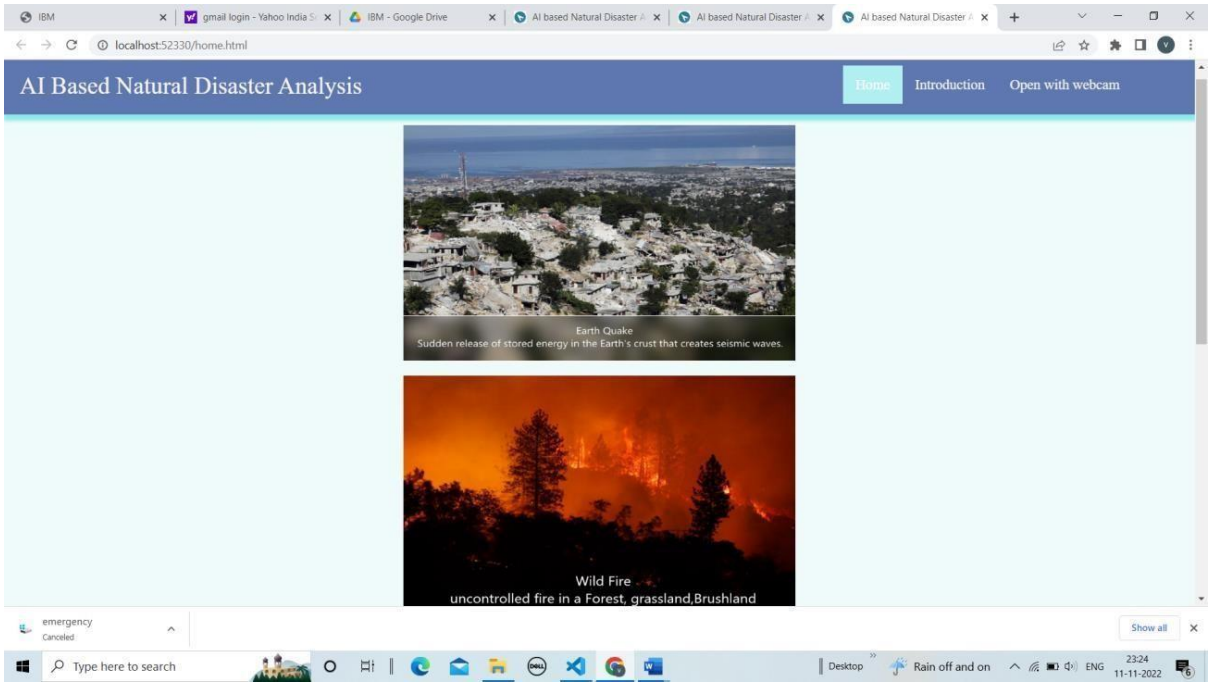
Home.html:

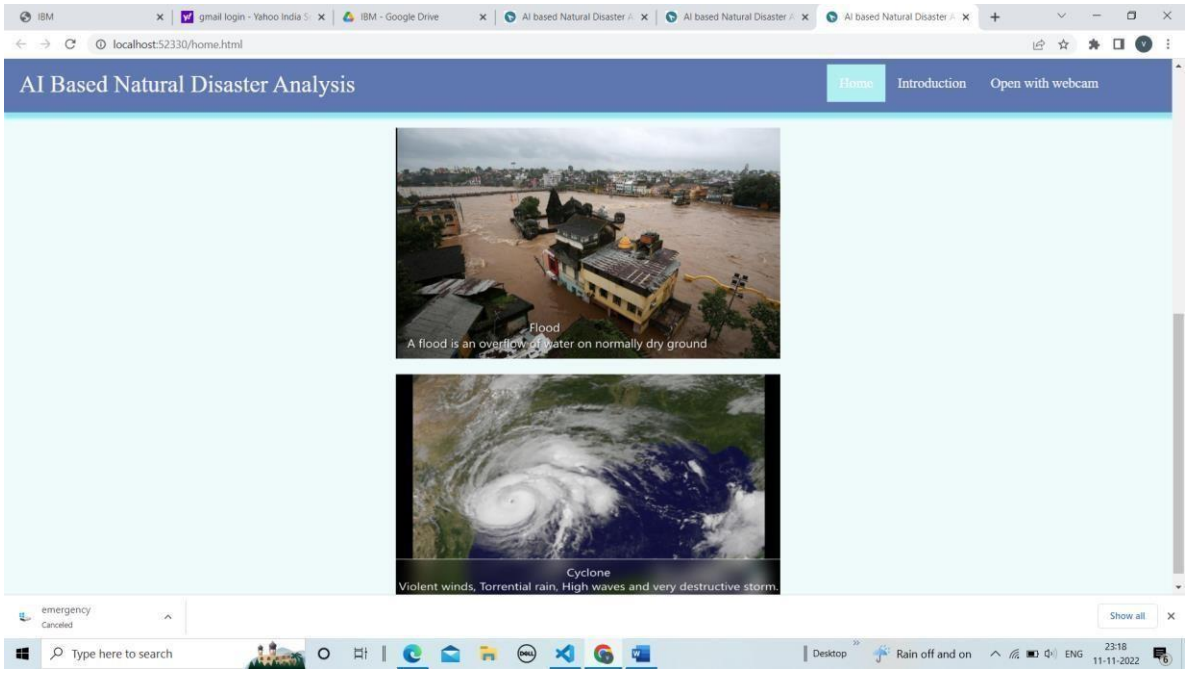
```
<!DOCTYPE html>
<html >
<head>
<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-
scale=1">

<title>AI based Natural Disaster Analysis</title>
<link href='https://fonts.googleapis.com/css?family=Pacifico'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300'
rel='stylesheet' type='text/css'>
<link
href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:3
00' rel='stylesheet' type='text/css'>
<!link rel="stylesheet" href="{{ url_for('static',
filename='css/style.css') }}">

<link
href='https://fonts.googleapis.com/css?family=Merriweather'
rel='stylesheet'>

<link href='https://fonts.googleapis.com/css?family=JosefinSans'
rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Montserrat'
rel='stylesheet'>
<style>
.header {
top: 0px;
margin:0px;
left: 0px;
right: 0px;
position: fixed;
background-color: hsl(222, 34%, 53%);
color: white;
box-shadow: 0px 8px 4px rgb(140, 234, 234);
overflow: hidden;
padding-left:20px;
font-family: 'Josefin Sans';
font-size: 2vw;
width: 100%;
height:10%;
```

```
text-align: center;
}
.topnav {
overflow: hidden;
background-color: rgb(179, 43, 43);
}
.topnav-right a {
float: left;
color: #f2f2f2;
text-align: center;
padding: 14px 16px;
text-decoration: none;
font-size: 18px;
}
.topnav-right a:hover {
background-color: #ddd;
color: black;
}
.topnav-right a.active {
background-color: #b1eff1;
color: white;
}

.topnav-right {
float: right;
padding-right:100px;
}
.login{
margin-top:-70px;
}
body {
background-color:rgb(239, 251, 251);
background-repeat: no-repeat;
background-size:cover;
background-position: 0px 0px;
}
.login
{
margin-top:100px;
}
form {background-color:rgb(229, 219, 227); border: 1px solid #2b2929;
margin-left:400px;margin-right:400px}
input[type=text],
input[type=email],input[type=number],input[type=password] {
width: 100%;
padding: 12px 20px;
display: inline-block;
margin-bottom:18px;
```

```
border: 1px solid rgb(252, 250, 252);
box-sizing: border-box;
}
button {
background-color: hsl(108, 63%, 56%);
color: rgb(253, 252, 252);
padding: 14px 20px;
margin-bottom:8px;
border: none;
cursor: pointer;
width: 100%;
font-weight:bold;
}
button:hover {
opacity: 0.8;
}
.cancelbtn {

width: auto;
padding: 10px 18px;
background-color: #f44336;
}
.imgcontainer {
text-align: center;
margin: 24px 0 12px 0;
}
img.avatar {
width: 50%;
border-radius: 50%;
}
.container {
padding: 16px;
}
span.psw {
float: right;
padding-top: 16px;
}
/* Change styles for span and cancel button on extra small
screens*/
@media screen and (max-width: 300px) {
span.psw {
display: block;
float: none;
}
.cancelbtn {
width: 100%;
}
}
```

```
img {
    display:block;
    margin-left:auto;
    margin-right:auto;
}
</style>
</head>
<body style="font-family:Montserrat;">
<div class="header">
<div style="width:50%; float:left; font-size:2vw; text-align:left;
color:rgb(249, 245, 245); padding-top:1%">AI Based Natural Disaster
Analysis</div><div
class="topnav-right" style="padding-top:0.5%;">
<a class="active" href="{{ url_for('index')}}">Home</a>
<a  href="{{ url_for('Login')}}">Introduction</a>
<a href="{{ url_for('register')}}">Open with webcam</a>
</div>
</div>

<div id="login" class="login"> </div>
<img src="earthQuakw.png"width="500"height="300"/>
<br>
<img src="wf (2).png"width="500"height="300"><br>
<img src="fp (2).png"width="500"height="300"><br>
<img src="cp (2).png"width="500"height="300">


</body>
</html>
```





Intro.html:

```
<!DOCTYPE html>
<html >
<head>
<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-
scale=1">
```

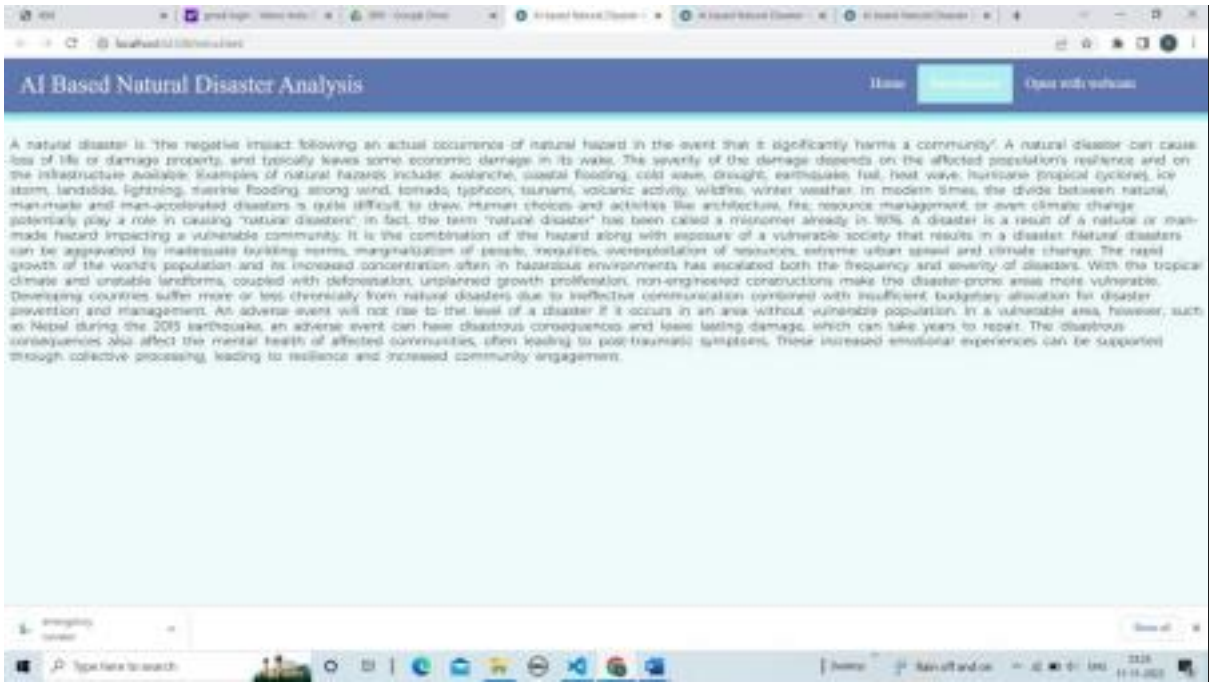
```
<title>AI based Natural Disaster Analysis</title>
<link href='https://fonts.googleapis.com/css?family=Pacifico'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300'
rel='stylesheet' type='text/css'>
<link
href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:3
00' rel='stylesheet' type='text/css'>
<!link rel="stylesheet" href="{{ url_for('static',
filename='css/style.css') }}">

<link
href='https://fonts.googleapis.com/css?family=Merriweather'
rel='stylesheet'>

<link href='https://fonts.googleapis.com/css?family=JosefinSans'
rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Montserrat'
rel='stylesheet'>
<style>
.header {
top: 0px;
margin:0px;
left: 0px;
right: 0px;
position: fixed;
background-color: hsl(222, 34%, 53%);
color: white;
box-shadow: 0px 8px 4px rgb(140, 234, 234);
overflow: hidden;
padding-left:20px;
font-family: 'Josefin Sans';
font-size: 2vw;
width: 100%;
height:10%;
text-align: center;
}
.topnav {
overflow: hidden;
background-color: rgb(179, 43, 43);
}
.topnav-right a {
float: left;
color: #f2f2f2;
text-align: center;
padding: 14px 16px;
```

```
text-decoration: none;
font-size: 18px;
}
.topnav-right a:hover {
background-color: #ddd;
color: black;
}
.topnav-right a.active {
background-color: #b1eff1;
color: white;
}

.topnav-right {
float: right;
padding-right:100px;
}
.login{
margin-top:-70px;
}
body {
background-color:rgb(239, 251, 251);
background-repeat: no-repeat;
background-size:cover;
background-position: 0px 0px;
}
.login
{
margin-top:100px;
}
form {background-color:rgb(229, 219, 227); border: 1px solid #2b2929;
margin-left:400px;margin-right:400px}
input[type=text],
input[type=email],input[type=number],input[type=password] {
width: 100%;
padding: 12px 20px;
display: inline-block;
margin-bottom:18px;
border: 1px solid rgb(252, 250, 252);
box-sizing: border-box;
}
button {
background-color: hsl(108, 63%, 56%);
color: rgb(253, 252, 252);
padding: 14px 20px;
margin-bottom:8px;
border: none;
cursor: pointer;
width: 100%;
```

```
font-weight:bold;
}
button:hover {
opacity: 0.8;
}
.cancelbtn {

width: auto;
padding: 10px 18px;
background-color: #f44336;
}
.imgcontainer {
text-align: center;
margin: 24px 0 12px 0;
}
img.avatar {
width: 50%;
border-radius: 50%;
}
.container {
padding: 16px;
}
span.psw {
float: right;
padding-top: 16px;
}
/* Change styles for span and cancel button on extra small
screens*/
@media screen and (max-width: 300px) {
span.psw {
display: block;
float: none;
}
.cancelbtn {
width: 100%;
}
}
img {
    display:block;
    margin-left:auto;
    margin-right:auto;
}

p {
  word-spacing: 5px;}

</style>
</head>
```

```
<body style="font-family:Montserrat;">
<div class="header">
<div style="width:50%; float:left; font-size:2vw; text-align:left;
color:rgb(249, 245, 245); padding-top:1%">AI Based Natural Disaster
Analysis</div><div
class="topnav-right" style="padding-top:0.5%;">
<a  href="{{ url_for('index')}}">Home</a>
<a  class="active"href="{{ url_for('Login')}}">Introduction</a>
<a href="{{ url_for('register')}}">Open with webcam</a>
</div>
</div>

<div id="login" class="login"> </div>

<p>
     A natural disaster is "the negative impact following an actual occurrence
of natural hazard in the event that it significantly harms a community". A
natural disaster can cause loss of life or damage property, and typically
leaves some economic damage in its wake. The severity of the damage depends on
the affected population's resilience and on the infrastructure available.
Examples of natural hazards include: avalanche, coastal flooding, cold wave,
drought, earthquake, hail, heat wave, hurricane (tropical cyclone), ice storm,
landslide, lightning, riverine flooding, strong wind, tornado, typhoon,
tsunami, volcanic activity, wildfire, winter weather.

   In modern times, the divide between natural, man-made and man-accelerated
disasters is quite difficult to draw. Human choices and activities like
architecture, fire, resource management or even climate change potentially
play a role in causing "natural disasters". In fact, the term "natural
disaster" has been called a misnomer already in 1976. A disaster is a result
of a natural or man-made hazard impacting a vulnerable community. It is the
combination of the hazard along with exposure of a vulnerable society that
results in a disaster.

    Natural disasters can be aggravated by inadequate building norms,
marginalization of people, inequities, overexploitation of resources, extreme
urban sprawl and climate change. The rapid growth of the world's population
and its increased concentration often in hazardous environments has escalated
both the frequency and severity of disasters. With the tropical climate and
unstable landforms, coupled with deforestation, unplanned growth
proliferation, non-engineered constructions make the disaster-prone areas more
vulnerable. Developing countries suffer more or less chronically from natural
disasters due to ineffective communication combined with insufficient
budgetary allocation for disaster prevention and management.

    An adverse event will not rise to the level of a disaster if it occurs in
an area without vulnerable population. In a vulnerable area, however, such as
Nepal during the 2015 earthquake, an adverse event can have disastrous
```

```
consequences and leave lasting damage, which can take years to repair. The
disastrous consequences also affect the mental health of affected communities,
often leading to post-traumatic symptoms. These increased emotional
experiences can be supported through collective processing, leading to
resilience and increased community engagement.
    </p>

</body>
</html>
```



Upload.html:

```
<!DOCTYPE html>
<html >
<head>
<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-
scale=1">

<title>AI based Natural Disaster Analysis</title>
<link href='https://fonts.googleapis.com/css?family=Pacifico'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo'
rel='stylesheet' type='text/css'>
```

```
<link href='https://fonts.googleapis.com/css?family=Hind:300'
rel='stylesheet' type='text/css'>
<link
href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:3
00' rel='stylesheet' type='text/css'>
<!link rel="stylesheet" href="{{ url_for('static',
filename='css/style.css') }}">

<link
href='https://fonts.googleapis.com/css?family=Merriweather'
rel='stylesheet'>

<link href='https://fonts.googleapis.com/css?family=JosefinSans'
rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Montserrat'
rel='stylesheet'>
<style>
.header {
top: 0px;
margin:0px;
left: 0px;
right: 0px;
position: fixed;
background-color: hsl(222, 34%, 53%);
color: white;
box-shadow: 0px 8px 4px rgb(140, 234, 234);
overflow: hidden;
padding-left:20px;
font-family: 'Josefin Sans';
font-size: 2vw;
width: 100%;
height:10%;
text-align: center;
}
.topnav {
overflow: hidden;
background-color: rgb(179, 43, 43);
}
.topnav-right a {
float: left;
color: #f2f2f2;
text-align: center;
padding: 14px 16px;
text-decoration: none;
font-size: 18px;
}
.topnav-right a:hover {
background-color: #ddd;
```

```
color: black;
}
.topnav-right a.active {
background-color: #b1eff1;
color: white;
}

.topnav-right {
float: right;
padding-right:100px;
}
.login{
margin-top:-70px;
}
body {
background-color:rgb(239, 251, 251);
background-repeat: no-repeat;
background-size:cover;
background-position: 0px 0px;
}
.login
{
margin-top:100px;
}
form {background-color:rgb(229, 219, 227); border: 1px solid #2b2929;
margin-left:400px;margin-right:400px}
input[type=text],
input[type=email],input[type=number],input[type=password] {
width: 100%;
padding: 12px 20px;
display: inline-block;
margin-bottom:18px;
border: 1px solid rgb(252, 250, 252);
box-sizing: border-box;
}
button {
background-color: hsl(108, 63%, 56%);
color: rgb(253, 252, 252);
padding: 14px 20px;
margin-bottom:8px;
border: none;
cursor: pointer;
width: 100%;
font-weight:bold;
}
button:hover {
opacity: 0.8;
}
```

```
.cancelbtn {

width: auto;
padding: 10px 18px;
background-color: #f44336;
}
.imgcontainer {
text-align: center;
margin: 24px 0 12px 0;
}
img.avatar {
width: 50%;
border-radius: 50%;
}
.container {
padding: 16px;
}
span.psw {
float: right;
padding-top: 16px;
}
/* Change styles for span and cancel button on extra small
screens*/
@media screen and (max-width: 300px) {
span.psw {
display: block;
float: none;
}
.cancelbtn {
width: 100%;
}
}
img {
    display:block;
    margin-left:auto;
    margin-right:auto;
}
p {text-align: center;}
p {
  word-spacing: 5px;}

</style>
</head>
<body style="font-family:Montserrat;">
<div class="header">
<div style="width:50%; float:left; font-size:2vw; text-align:left;
color:rgb(249, 245, 245); padding-top:1%">AI Based Natural Disaster
Analysis</div><div
```

```
class="topnav-right" style="padding-top:0.5%;">
<a  href="{{ url_for('index')}}">Home</a>
<a href="{{ url_for('Login')}}">Introduction</a>
<a  class="active" href="{{ url_for('register')}}">Open with webcam</a>
</div>
</div>

<div id="login" class="login"> </div>

<img src="emergency.png"width="700"height="600"><br>

</body>
</html>
```

Flask(application.py)

```
# import the necessary packages
from flask import Flask, render_template, request, redirect, url_for
# Flask-It is our framework which we are going to use to run/serve our
application.
# request-for accessing file which was uploaded by the user on our
application.
#import operator
import cv2  # opencv library
from tensorflow.keras.models import load_model  # to load our trained
model
import numpy as np
#import os
```

```
from werkzeug.utils import secure_filename
#from playsound import playsound
#from gtts import gTTS

app = Flask(__name__, template_folder="templates")  # initializing a
flask app
# Loading the model
model = load_model('disaster.h5')
print("Loaded model from disk")


# app=Flask(__name__,template_folder="templates")
@app.route('/', methods=['GET'])
def index():
    return render_template('home.html')


@app.route('/home', methods=['GET'])
def home():
    return render_template('home.html')


@app.route('/intro', methods=['GET'])
def about():
    return render_template('intro.html')


@app.route('/upload', methods=['GET', 'POST'])
def predict():

    # Get a reference to webcam #0 (the default one)

    print("[INFO] starting video stream...")
    vs = cv2.VideoCapture(0)
    #writer = None
    (W, H) = (None, None)

    # loop over frames from the video file stream
    while True:
        # read the next frame from the file
        (grabbed, frame) = vs.read()

        # if the frame was not grabbed, then we have reached the end
        # of the stream
        if not grabbed:
            break

            # if the frame dimensions are empty, grab them
        if W is None or H is None:
            (H, W) = frame.shape[:2]

            # clone the output frame, then convert it from BGR to RGB
            # ordering and resize the frame to a fixed 64x64
        output = frame.copy()
        # print("apple")
        frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        frame = cv2.resize(frame, (64, 64))
        #frame = frame.astype("float32")
```

```
        x = np.expand_dims(frame, axis=0)
        result = np.argmax(model.predict(x), axis=-1)
        index = ['Cyclone', 'Earthquake', 'Flood', 'Wildfire']
        result = str(index[result[0]])
        # print(result)
        # result=result.tolist()

        cv2.putText(output, "activity: {}".format(result), (10, 120),
cv2.FONT_HERSHEY_PLAIN,
                    1, (0, 255, 255), 1)
        #playaudio("Emergency it is a disaster")
        cv2.imshow("Output", output)
        key = cv2.waitKey(1) & 0xFF

        # if the `q` key was pressed, break from the loop
        if key == ord("q"):
            break

    # release the file pointers
    print("[INFO] cleaning up...")
    vs.release()
    cv2.destroyAllWindows()
    return render_template("upload.html")


@app.route('/file', methods=['POST', 'GET'])
def video():
    if request.method == 'POST':
        uploaded_file = request.files['file1']
        if uploaded_file.filename != '':
            vid_name = str(uploaded_file.filename)
            print(vid_name + "Uploaded_Succesfully")
            uploaded_file.save(uploaded_file.filename)
            vs = cv2.VideoCapture(vid_name)
            if (vs.isOpened() == False):
                print("Error opening video stream or file")

            (W, H) = (None, None)
            while True:
                (grabbed, frame) = vs.read()
                if not grabbed:
                    break
                if W is None or H is None:
                    (H, W) = frame.shape[:2]
                output = frame.copy()
                frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
                frame = cv2.resize(frame, (64, 64))
                x = np.expand_dims(frame, axis=0)
                result = np.argmax(model.predict(x), axis=-1)
                index = ['Cyclone', 'Earthquake', 'Flood', 'Wildfire']
                result = str(index[result[0]])
                cv2.putText(output, "activity: {}".format(
                    result), (10, 120), cv2.FONT_HERSHEY_PLAIN, 1, (0,
255, 255), 1)
                cv2.imshow("Output", output)
                key = cv2.waitKey(1) & 0xFF
                if key == ord("q"):
                    break
```

```
            print("[INFO] cleaning up...")
            vs.release()
            cv2.destroyAllWindows()
    return render_template("file.html")


@app.route('/image', methods=['POST', 'GET'])
def image():
    resulttext = ''
    if request.method == 'POST':
        uploaded_file = request.files['imgfile']
        if uploaded_file.filename != '':
            img_name = str(uploaded_file.filename)
            print(img_name + "Uploaded Succesfully")
            uploaded_file.save(uploaded_file.filename)
            from tensorflow.keras.models import load_model
            from keras.preprocessing import image
            model = load_model("disaster.h5")  # loading the model for
testing
            img = image.load_img(img_name, grayscale=False,
                                 target_size=(64, 64))  # loading of
the image
            x = image.img_to_array(img)  # image to array
            x = np.expand_dims(x, axis=0)  # changing the shape
            pred = model.predict_classes(x)  # predicting the classes
            index = ['Cyclone', 'Earthquake', 'Flood', 'Wildfire']
            result = index[pred[0]]
            resulttext = result
    return render_template('image.html', result_text=resulttext)


if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8000, debug=True)
```

output:

AI based Natural disaster analysis.ipynb

AI based Natural disaster analysis

Importing Neccessary Libraries

```
[ ] import numpy as np#used for numerical analysis
    import tensorflow #open source used for both ML and DL for computation
    from tensorflow.keras.models import Sequential #it is a plain stack of layers
    from tensorflow.keras import layers #A layer consists of a tensor-in tensor-out computation function
    #Dense layer is the regular deeply connected neural network layer
    from tensorflow.keras.layers import Dense,Flatten
    #Faltten-used fot flattening the input or change the dimension
    from tensorflow.keras.layers import Conv2D,MaxPooling2D #Convolutional layer
    #MaxPooling2D-for downsampling the image
    from keras.preprocessing.image import ImageDataGenerator
```

```
Using TensorFlow backend.
```

Loading our data and performing data agumentation

```
[ ] #performing data agumentation to train data
    x_train = train_datagen.flow_from_directory(r'E:\SB1\dataset\dataset\dataset\test_set',target_size=(64, 64),batch_size=5,
                                                color_mode='rgb',class_mode='categorical')
    #performing data agumentation to test data
    x_test = test_datagen.flow_from_directory(r'E:\SB1\dataset\dataset\dataset\test_set',target_size=(64, 64),batch_size=5,
                                              color_mode='rgb',class_mode='categorical')
```

```
Found 198 images belonging to 4 classes.
Found 198 images belonging to 4 classes.
```

```
print(x_train.class_indices)#checking the number of classes
```

```
{'Cyclone': 0, 'Earthquake': 1, 'Flood': 2, 'Wildfire': 3}
```

+ Code + Text

```
[ ] print(x_test.class_indices)#checking the number of classes
```

```
{'Cyclone': 0, 'Earthquake': 1, 'Flood': 2, 'Wildfire': 3}
```

Creating the model

```
# Initializing the CNN
classifier = Sequential()

# First convolution layer and poolingo
classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))
classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation='relu'))
# Second convolution layer and pooling
classifier.add(Conv2D(32, (3, 3), activation='relu'))
# input_shape is going to be the pooled feature maps from the previous convolution layer
classifier.add(MaxPooling2D(pool_size=(2, 2)))
classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation='relu'))

# Flattening the layers
classifier.add(Flatten())

# Adding a fully connected layer
classifier.add(Dense(units=128, activation='relu'))
classifier.add(Dense(units=4, activation='softmax')) # softmax for more than 2
```

Fitting the model

```
[ ] classifier.fit_generator(
            generator=x_train,steps_per_epoch = len(x_train),
            epochs=40, validation_data=x_test,validation_steps = len(x_test))# No of images in test set
```

```
40/40 [==============================] - 9s 239ms/step - loss: 0.7445 - accuracy: 0.7266 - val_loss: 0.6234 - val_accuracy: 0.7172
Epoch 13/40
40/40 [==============================] - 9s 239ms/step - loss: 0.5752 - accuracy: 0.7508 - val_loss: 0.5389 - val_accuracy: 0.7980
Epoch 14/40
40/40 [==============================] - 10s 242ms/step - loss: 0.6582 - accuracy: 0.7428 - val_loss: 0.4447 - val_accuracy: 0.8283
Epoch 15/40
40/40 [==============================] - 9s 240ms/step - loss: 0.5318 - accuracy: 0.7766 - val_loss: 0.4859 - val_accuracy: 0.8131
Epoch 16/40
40/40 [==============================] - 9s 240ms/step - loss: 0.4472 - accuracy: 0.8269 - val_loss: 0.6708 - val_accuracy: 0.7273
Epoch 17/40
40/40 [==============================] - 10s 246ms/step - loss: 0.5900 - accuracy: 0.7400 - val_loss: 0.6847 - val_accuracy: 0.7525
Epoch 18/40
40/40 [==============================] - 10s 249ms/step - loss: 0.5226 - accuracy: 0.8148 - val_loss: 0.8422 - val_accuracy: 0.7222
Epoch 19/40
40/40 [==============================] - 11s 277ms/step - loss: 0.5587 - accuracy: 0.8253 - val_loss: 0.4669 - val_accuracy: 0.8081
Epoch 20/40
```

Saving our model

```
[ ] # Save the model
    classifier.save('disaster_f.h5')
```

```
[ ] model_json = classifier.to_json()
    with open("model-bw.json", "w") as json_file:
        json_file.write(model_json)
```

```
[ ]
```

Predicting our results

```
from tensorflow.keras.models import load_model
from keras.preprocessing import image
#model = load_model("disaster_f.h5") #loading the model for testing
```

c_check_earthquake.py:

```
def check_earthquakes_location(lat_out=float,lon_out=float):

    try:

        TARGET_REQ_URL = "https://www.emsc-csem.org/service/rss/rss.php?typ=emsc"

        REQ_TARGET = requests.get(TARGET_REQ_URL).text
        SOUP_TARGET = BeautifulSoup(REQ_TARGET,"html.parser")

        FIND_ALL_IT = SOUP_TARGET.find_all("item")

        checking_value = 0

        print("\n")
        time.sleep(1.2)
        print("CONNECTED PORTAL I")


        for x_loop in FIND_ALL_IT:

            TITLE_OUT = x_loop.find("title")
            LAT_OUT = x_loop.find("geo:lat")
            LON_OUT = x_loop.find("geo:long")
```

C_earthquake_g.py:

```
def get_earthquake(count_search=int):

    SOURCE_URL ='https://ds.iris.edu/seismon/eventlist/index.phtml'

    try:

        MAIN_URL_REQ = requests.get(SOURCE_URL).text
        MAIN_SOUP_URL = BeautifulSoup(MAIN_URL_REQ,"html.parser")
        PARAMS_ALL_GET = MAIN_SOUP_URL.find_all("table",class_="tablesorter")

        i_count_stop = 0

        for X_DETAIL in PARAMS_ALL_GET:

            DETAIL_TR_ALL = X_DETAIL.find_all("tr")

            for x_d in DETAIL_TR_ALL:

                LIST_DETAIL_ALL = x_d.text.replace("\n",",").split(",")
                i_count_stop += 1

                if 1 < i_count_stop < count_search:
```

c_alternative_earthquake.py:

```
def check_earthquakes_location(lat_out=float,lon_out=float):

    try:

        TARGET_REQ_URL = "https://www.emsc-csem.org/service/rss/rss.php?typ=emsc"

        REQ_TARGET = requests.get(TARGET_REQ_URL).text
        SOUP_TARGET = BeautifulSoup(REQ_TARGET,"html.parser")

        FIND_ALL_IT = SOUP_TARGET.find_all("item")

        checking_value = 0

        print("\n")
        time.sleep(1.2)
        print("CONNECTED PORTAL I")


        for x_loop in FIND_ALL_IT:

            TITLE_OUT = x_loop.find("title")
            LAT_OUT = x_loop.find("geo:lat")
            LON_OUT = x_loop.find("geo:long")
```

C_flood.py:

```
def get_flood(count_search=int):

    try:

        GDACS_TARGET = requests.get("https://www.gdacs.org/default.aspx").text
        SOUP_GDACS = BeautifulSoup(GDACS_TARGET,"html.parser")
        ALL_F_DISASTER = SOUP_GDACS.find_all("div",id="mainListFl")

        CONTROL_VALUE_LIST = []

        i_count_stop = 0


        for x_att in ALL_F_DISASTER:

            ALERT_DETAIL_LINK = x_att.find_all("a")

            for x_detail_link in ALERT_DETAIL_LINK:

                LINK_AFTER_SITE = str(x_detail_link.get("href"))

                SUB_TARGET = requests.get(LINK_AFTER_SITE).text
                SOUP_GDACS_FUNCTION = BeautifulSoup(SUB_TARGET,"html.parser")
```

c_help.py:

```
def how_to_use():

    try:

        MY_TEXT = "ISC INITIATIVE"

        MY_FONT = ImageFont.truetype("verdanab.ttf", 11)
        MY_SIZE = MY_FONT.getsize(MY_TEXT)

        MY_IMG = Image.new("1",MY_SIZE,"black")
        DRAW_FUNC = ImageDraw.Draw(MY_IMG)
        DRAW_FUNC.text((0, 0), MY_TEXT, "white", font=MY_FONT)

        PIX_RES = np.array(MY_IMG, dtype=np.uint8)
        CHAR_RES = np.array([' ','#'], dtype="U1")[PIX_RES]

        STR_RES = CHAR_RES.view('U' + str(CHAR_RES.shape[1])).flatten()
        print("\n".join(STR_RES))

    except:

        pass
```

c_import.py

```
"""
(cc) Creative Commons / 2020-2021 ISCI - LAB DEVELOPERS
We are an initiative that conducts studies in the field of Space Science, publishes projects and reports, offers analytical perspectives and data ana
We believe that science changes the future.
initiative.isc@protonmail.com
initiative.isc@tutanota.com
"""

from __future__ import print_function

try:

    from PIL import Image, ImageDraw, ImageFont
    from optparse import OptionParser
    import requests
    from bs4 import BeautifulSoup
    import sys
    import time
    import numpy as np
    import warnings

    warnings.filterwarnings(action="ignore",message="CHECK PYTHON VERSION")
    warnings.filterwarnings(action="ignore",message="ALREADY IMPORTED",category=UserWarning)
```

C_local_alert:

```
def get_local_based(search_parameters=str,count_search=int):

    TARGET_URL = f"https://severeweather.wmo.int/{search_parameters}/"

    try:

        TAR_REQ = requests.get(TARGET_URL).text
        BS_REQ = BeautifulSoup(TAR_REQ,"html.parser")

        Area_ALL = BS_REQ.find_all("area")

        i_count_stop = 0

        for x_loop_area in Area_ALL:

            HREF_ALL_AREA_PATH = x_loop_area.get("href")
            REP_DOT_RAIN = HREF_ALL_AREA_PATH.replace("./","")
            ALL_PATH_RAIN = TARGET_URL + REP_DOT_RAIN

            NEW_TAR_REQ = requests.get(ALL_PATH_RAIN).text
            BS_NEW_TAR = BeautifulSoup(NEW_TAR_REQ,"html.parser")
            AREA_NEW_ALL = BS_NEW_TAR.find_all("area")
```


C_nasaoent.py:

```
def get_nasa_eonet(count_search=int):

    TEST_SPEC_TARGET_URL = "https://eonet.sci.gsfc.nasa.gov/api/v3/events"

    try:

        READ_URL = requests.get(TEST_SPEC_TARGET_URL)
        READ_JSON = READ_URL.json()

        EVENTS_JSON = READ_JSON["events"]


        for x_range in range(count_search):


            EVENT_TITLE = EVENTS_JSON[x_range]["title"]
            EVENT_DATE = EVENTS_JSON[x_range]["geometry"][0]["date"]
            EVENTS_LAT = EVENTS_JSON[x_range]["geometry"][0]["coordinates"][1]
            EVENTS_LON = EVENTS_JSON[x_range]["geometry"][0]["coordinates"][0]

            time.sleep(0.8)
            print("\n")
            print("TITLE: ",EVENT_TITLE)
```

c_seismic.py:

```
def get_seismic_data(count_search=int):

    TARGET_REQ_URL = f"https://www.seismicportal.eu/mtws/api/search?&format=json&downloadAsFile=false&orderby=tim

    try:

        READ_URL = requests.get(TARGET_REQ_URL)
        READ_JSON = READ_URL.json()

        for x_num in range(len(READ_JSON)):

            NEW_JSON = READ_JSON[x_num]

            time.sleep(0.8)
            print("\n")
            print("REGION: ",NEW_JSON["ev_region"])
            print("LATITUDE: ",NEW_JSON["ev_latitude"])
            print("LONGITUDE: ",NEW_JSON["ev_longitude"])
            print("DEPTH: ",NEW_JSON["ev_depth"])
            print("MAGNITUDE VALUE: ",NEW_JSON["ev_mag_value"])
            print("MAGNITUDE TYPE: ",NEW_JSON["ev_mag_type"])
            print("EVENT TIME: ",NEW_JSON["ev_event_time"])
            print("FULL COUNT: ",NEW_JSON["full_count"])
```

C_valcano_g:

```
def get_volcano(count_search=int):

    try:

        GDACS_TARGET = requests.get("https://www.gdacs.org/default.aspx").text
        SOUP_GDACS = BeautifulSoup(GDACS_TARGET,"html.parser")
        ALL_V_DISASTER = SOUP_GDACS.find_all("div",id="mainListVo")
        CONTROL_VALUE_LIST = []
        i_count_stop = 0


        for x_att in ALL_V_DISASTER:

            ALERT_DETAIL_LINK = x_att.find_all("a")

            for x_detail_link in ALERT_DETAIL_LINK:

                LINK_AFTER_SITE = str(x_detail_link.get("href"))

                SUB_TARGET = requests.get(LINK_AFTER_SITE).text
                SOUP_TARGET = BeautifulSoup(SUB_TARGET,"html.parser")
                SUB_TARGET_DIV = SOUP_TARGET.find_all("div",id="alert_summary_left")
```

C_valcano_alternative:

```
def get_alternative_volcano():

    TARGET_REQ_URL = "https://volcano.si.edu/news/WeeklyVolcanoRSS.xml"

    try:

        REQ_TARGET = requests.get(TARGET_REQ_URL).text
        SOUP_TARGET = BeautifulSoup(REQ_TARGET,"html.parser")

        FIND_ALL_ITE = SOUP_TARGET.find_all("item")


        for x_loop in FIND_ALL_ITE:

            TITLE_OUT = x_loop.find("title")
            DES_OUT = x_loop.find("description")
            COOR_OUT = x_loop.find("georss:point")

            time.sleep(0.8)
            print("\n")
            print("TITLE: ",TITLE_OUT.text)
            print("DESCRIPTION: ",DES_OUT.text.replace("<p>","").replace("</p>",""))
```

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-12683-1659457710

DEMO LINK:

https://drive.google.com/file/d/1i7YupO6cx28sGdAE0eS6hLZmGg6oHenV/view?usp=share_link