CMR UNIVERSITY
Private University Established in Karnataka State by Act No. 45 of 2013

SCHOOL OF ENGINEERING AND TECHNOLOGY

CAPSTONE REPORT

On

"Mental Health Analysis ChatBot INSPIRED"

Submitted in partial fulfillment of the requirements for the award of degree in

Bachelor of Technology

in

Computer Science and Engineering

Submitted by:

ARYAN DEVIDAS MIDDLAGAJANI - 16UG08006
MELVIN DERRICK - 16UG08021
SOWJANYA - 16UG08057
SRIKATH S - 16UG08059

Under the Guidance of:

PROF. SHRUTI HEGDE
Assistant Professor
Dept. of CSE, SOET

Department of Computer Science and Engineering

Off Hennur - Bagalur Main Road,

Near Kempegowda International Airport, Chagalahatti,

Bangalore, Karnataka-562149

2019-2020

CMR UNIVERSITY

Private University Established in Karnataka State by Act No. 45 of 2013

SCHOOL OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

CERTIFICATE

Certificate that the CAPSTONE work, entitled **"Mental Health Analysis Chatbot INSPIRED"**, Carried out by **Mr. Aryan Devidas Middlagajani, USN: 16UG08006, Mr. Melvin Derrick, USN: 16UG08021, Ms. Sowjanya, USN: 16UG08057, Mr. Srikanth S, USN: 16UG08059**, a bonafied student of CMR UNIVERSITY in partial fulfillment for the award of Bachelor of Technology in Computer Science Engineering, Bengaluru for the academic year 2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library.

Signature of Guide	Signature of HOD	Signature of Dean
……	..……	.....……
Dept. of CSE SoET, CMRU, Bangalore	Dept. of CSE SoET, CMRU, Bangalore	SoET, CMRU, Bangalore

External Viva:

Name of the Examiners:	Signature with Date:
1 …….	……
2 ……. .	……

DECLARATION

We, **Aryan Devidas Middlagajani (Reg. No. 16UG08006), Melvin Derrick (Reg. No. 16UG08021), Sowjanya (Reg. No. 16UG08057), Srikanth S (Reg. No. 16UG08059)** student of 8th semester B.Tech. Computer Science and Engineering, School of Engineering and Technology, Bangalore, hereby declare that the CAPSTONE work entitle **"Mental Health Analysis Chatbot INSPIRED"** has been carried out by us under the guidance of **Prof. Shruti Hegde**, Department of Computer Science and Engineering, School of Engineering and Technology. This report is submitted in partial fulfillment of the requirement for award of Bachelor of Technology in Computer Science and Engineering by CMR University, Bangalore during the academic year 2019-2020. The matter embodied in the dissertation has not been submitted previously by anybody for the award of any degree or diploma to any other university.

Place: Bangalore

Date:

ARYAN D M
(16UG08006)

MELVIN DERRICK
(16UG08021)

SOWJANYA
(16UG08057)

SRIKANTH S
(16UG08059)

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of this project would be incomplete without the mention of the people who made it possible, without whose constant guidance and encouragement would have made efforts go in vain.

We consider ourselves privileged to express gratitude and respect towards all those who guided us through the completion of the project. We express our heartfelt sincere gratitude to **Dr. Jayaprasad M, Dean**, School of Engineering and Technology, CMR University for his support.

We would like to express our thanks to **Dr. Arun Biradar, Professor and Head**, Department of Computer Science and Engineering, School of Engineering and Technology, CMR University, Bangalore, for his encouragement that motivated us for the successful completion of Project work.

We express our thanks to our Project Guide **Prof. Shruti Hegde, Assistant Professor**, Department of Computer Science and Engineering, School of Engineering and Technology, CMR University for her constant support.

We would like to thank all the professors and staff of Computer Science and Engineering Department for their co-operation and timely guidance.

ARYAN D M
(16UG08006)

MELVIN DERRICK
(16UG08021)

SOWJANYA
(16UG08057)

SRIKANTH S
(16UG08059)

ABSTRACT

Over the last decade, there has been an explosion of digital interventions that aim to either supplement or replace face-to-face mental health services. More recently, a number of automated conversational agents have also been made available, which respond to users in ways that mirror a real-life interaction. What are the social and ethical concerns that arise from these advances? In this Project, we discuss, from a young person's perspective, the strengths and limitations of using chat bots in mental health support. We also outline what we consider to be minimum ethical standards for these platforms, including issues surrounding privacy and confidentiality, efficacy, and safety, and review three existing platforms (Woebot, Joy, and Wysa) according to our proposed framework. Mental health is nowadays a topic which is most frequently discussed when it comes to research but least frequently discussed when it comes to the personal life. The well-being of a person is the measure of mental health. The increasing use of technology will lead to a lifestyle of less physical work. Also, the constant pressure on humans in any industry will make more vulnerable to mental disorder. These vulnerabilities consist of peer pressure, anxiety attack, depression, and many more. Based on their answers the result is derived. Different machine learning techniques are used to get the results. It is our hope that this project inspire ethically responsible practice in digital mental health.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1Description

This project Mental Health Analysis is using Reactjs. It's a big deal. According to the World Health Organization research, more than 300 million people are suffering from depression alone, not to mention other types of mental issues. This kind of thing is almost inevitable among professionals of any field. And while the need for mental health services is getting higher, its availability is unable to catch up. In this environment, the traditional means of psychological relief and stress management aren't efficient enough. The rapid development of natural language processing and conversational interfaces has enabled a more progressive way of dealing with mental health problems, stress management, and psychological relief. A mental health chatbot is a type of a conversational interface application designed to have a conversation with the patient regarding his mental wellbeing; Provide instant 24/7 available chat; Deliver detached statistics for the patient to self-regulate his mental state; Give users basic recommendations on how to improve a patient's mental wellbeing. Mental health chatbots are designed to maintain a conversation, not to lead it. In a way, using a mental health chatbot resembles practicing tennis against a wall. The general functional framework of mental health chatbots is based on Cognitive Behavioral Therapy methodology. CBT is a form of talking therapy designed to manage mental health states by rearranging the way the patient perceives it, i.e., making negative thoughts positive. Reactjs is a declarative, efficient, and flexible JavaScript library (A JavaScript library is a library of pre-written JavaScript which allows for easier development of JavaScript-based applications) for building user interfaces. A chatbot is a bot type that simulates human conversation (a chat). Its main role is to automate the process of interaction with website visitors and social media followers. Think of the chatbot as a robot that answers questions round-the-clock. Chatbots interact with humans via a (live) chat interface or SMS. Live chat presence on the website is indicated by the chat widget. AI-powered chatbots leverage Machine Learning and Natural Language Processing to deliver more natural conversation AI-powered chatbots learn as they talk. The more input and feedback they receive the better experience in chatting they build up. The process is similar to human reasoning.

The primary goal of this project is to extract patterns from a Mental health counselling dataset, and then build a model based on these extracted patterns, in order to predict the mental health issue by using classification data mining algorithms. Then process it using AI chatbot natural language processing and then to build a user interface.

1.2Background and Problem statement

The last two decades have seen a proliferation of Internet- delivered and mobile mental health interventions both within research settings and on the market. Unsupported or unguided interventions offer fully automated self-help services, whereas supported or guided interventions provide additional human support by a remote coach or clinician. Research studies on the clinical benefits of these interventions for the treatment of anxiety and depression have shown mixed results, but most have denounced the poor adherence that characterizes digital mental health interventions. More recently, a number of engaging, fully automated conversational agents, or chatbots, have been made available. These text-based platforms, easily accessible via a mobile app or Facebook Messenger, are designed in a way that increases adherence, because they engage with users in ways that resemble human real-life interactions. The main goal of this project is to extract patterns from a mental health counselling dataset, and then build a model based on these extracted patterns, in order to predict the mental health issue by using classification data mining algorithms. Then process it using AI chatbot natural language processing and then to build a user interface. We believe that perspectives of young people are essential to such discussions, because mental illness accounts for a large portion of the disease burden in younger populations, yet young people's mental health needs are largely unmet, due to under-investment in child and adolescents' mental health, lack of evidence-based treatments, and poorly targeted or badly implemented interventions. At the same time, young people are the largest consumer of the digital world, and as such they constitute one of the key target groups for digital interventions. Mental health issues such as stress, anxiety, and depression are among the health information most searched for by teenagers online, and online content often motivates young people to change their health behaviors· These data suggest that digital mental health interventions are likely to affect young people in a very important dimension of their lives. At a time where young people are seen as simultaneously in need of protection from the risks of the digital world, and as 'media-savvy' individuals who are much more confident and competent in the use of digital devices than adults, we believe that

young people's first-hand experience of mental health chatbots should play a role in shaping the normative debate around e-Mental Health.

1.3Objective

- The primary goal of this project is to extract patterns from a mental health counselling dataset, and then build a model based on these extracted patterns, in order to predict the mental health issue by using classification data mining algorithms.
- Process it using AI chatbot natural language processing and then to build a user interface.
- Mental health issues such as stress, anxiety, and depression are among the health information most searched for by teenagers online, and online content often motivates young people to change their health behaviors.
- These data suggest that digital mental health interventions are likely to affect young people in a very important dimension of their lives.
- Users receive reports that might help them gain insight into their own patterns, as well as targeted therapy exercises, including reframing one's thoughts, mindful breathing, and motivational interviewing, in the form of text, games, or video clips.
- According to their developers, chatbots can be used in settings with Internet connection, because they are designed to use limited phone data. Finally, because they are commercially accessible rather than available in academic research settings only, chatbots are likely to be more sustainable than other e-Mental Health services
- Chatbots may offer great potential for providing people with useful help for mental health difficulties. There are, however, several ethical issues associated with this potential. While testing the chatbots, we were particularly concerned about matters related to who has access to users' personal information and conversations; whether the digital support provided is evidence-based; and how automated bots protect users' safety in emergency situations.
- For their potential not to be compromised, we believe that automated bots should meet a set of minimum ethical standards concerning privacy and confidentiality, efficacy, and safety.

CHAPTER 2

LITERATURE SURVEY

Lot of work has been done with regards to extracting important data, to learn about the domain of the project. About machine learning, about Reactjs and Chatbot including AI chatbot and also about the Natural language processing. My project aim is to gather all the mental health information from multiple sources or multiple counselling and applying different classification algorithms, which could give best prediction results. I have taken the reference of the work listed below in order to do my analysis.

1. Paper Title: about Can your phone be your therapist? Young people ethical perspectives on the use of fully automated conversational agent (chatbots) in mental health support.

Authors name: Kira Kretzschmar, Holly Tyroll, Gabriela Pavarini, Arianna Manzini and Ilina Singh

The paper by Kira Kretzschmar, Holly Tyroll, Gabriela Pavarini, Arianna Manzini and Ilina Singh [1] discussed about Can your phone be your therapist? Young people ethical perspectives on the use of fully automated conversational agent (chatbots) in mental health support. This article arguing that bots like Woebot, Wysa, and Joy might have great potential to provide help to people struggling with mental health problems. Chatbots are widely available and accessible to anyone with a smartphone and Internet access, and they may be perceived to be less stigmatizing than formal mental health support. Over the last decade, there has been an explosion of digital interventions that aim to either supplement or replace face-to-face mental health services. More recently, a number of automated conversational agents have also been made available, which respond to users in ways that mirror a real-life interaction. What are the social and ethical concerns that arise from these advances? In this article, we discuss, from a young person's perspective, the strengths and limitations of using chatbots in mental health support. We also outline what we consider to be minimum ethical standards for these platforms, including issues surrounding privacy and confidentiality, efficacy, and safety, and review three existing platforms (Woebot, Joy, and Wysa) according to our proposed framework. It is our hope that this article will stimulate ethical debate among app developers, practitioners, young people, and other stakeholders, and inspire ethically responsible practice in digital mental health.

2. **Paper Title: Prediction of Mental Disorder for employees in IT Industry.**

Authors name: Sandhya P, Mahek Kantesaria

Sandhya P, Mahek Kantesaria [2] their paper stated about the Prediction of Mental Disorder for employees in IT Industry. In this paper they discussed about the implementation of different machine learning algorithms like logistic regression, Knn, Decision tree, Random forest, Bagging, Boosting, Neural network. Mental health is nowadays a topic which is most frequently discussed when it comes to research but least frequently discussed when it comes to the personal life. The well-being of a person is the measure of mental health. The increasing use of technology will lead to a lifestyle of less physical work. Also, the constant pressure on an employee in any industry will make more vulnerable to mental disorder. These vulnerabilities consist of peer pressure, anxiety attack, depression, and many more. Here we have taken the dataset of the questionnaires which were asked to an IT industry employee. Based on their answers the result is derived. Here output will be that the person needs an attention or not. Different machine learning techniques are used to get the results. This prediction also tells us that it is very important for an IT employee to get the regular mental health check up to tract their health. The employers should have a medical service provided in their company and they should also give benefits for the affected employees.

3. Paper Title: Machine learning for precision psychiatry

Authors name: Prof. Danilo Bzdok, M.D., Ph.D. & Prof. Andreas Meyer-Lindenberg, M.D

The paper by Prof. Danilo Bzdok, M.D., Ph.D. & Prof. Andreas Meyer-Lindenberg, M.D [3] discussed about the concept of Machine learning for precision psychiatry. The nature of mental illness remains a conundrum. Traditional disease categories are increasingly suspected to mis-represent the causes underlying mental disturbance. Yet, psychiatrists and investigators now have an unprecedented opportunity to benefit from complex patterns in brain, behavior, and genes using methods from machine learning (e.g., support vector machines, modern neural-network algorithms, cross-validation procedures). Combining these analysis techniques with a wealth of data from consortia and repositories has the potential to advance a biologically grounded re-definition of major psychiatric disorders. Within the next 10-20 years, incoming patients could be

stratified into distinct biological subgroups that cut across classical diagnostic boundaries. In a new era of evidence-based psychiatry tailored to single patients, objectively measurable endophenotypes could allow for individualized prediction of early diagnosis, treatment selection, and dosage adjustment to reduce the burden of disease. This primer aims to introduce clinicians and researchers to the opportunities and challenges in bringing machine intelligence into psychiatric practice.

4. Paper Title: Predicting mental health early and precisely has major implications for clinical management and practice, and ultimately life expectancy.

Authors name: Proff. Carola-Bibiane Schonlieb, Proff. Terry Lyons and Proff. Peter Tino

Proff. Carola-Bibiane Schonlieb, Proff. Terry Lyons and Proff. Peter Tino [4] proposed about the Predicting mental health early and precisely has major implications for clinical management and practice, and ultimately life expectancy. This project will use machine learning techniques to produce robust modelling tools that aim to improve the precision of clinical practice in mental health. Machine learning will be used to predict and classify disease risk at an individual level (for dementia, anxiety, depression, and others) and to determine the interactive factors that influence mental health across people's lifespans (e.g. genetics, cognition, demographics).

5. Paper Title: Behavioral Modeling for Mental Health using Machine Learning Algorithms

Authors name: M. Srividya & S. Mohanavalli1 & N. Bhalaji

The paper by M. Srividya & S. Mohanavalli1 & N. Bhalaji [5] discussed about the Behavioral Modeling for Mental Health using Machine Learning Algorithms. The aim of this research work is to identify the individuals who are mentally distressed in the target population. These individuals need special attention in order to mentally balanced population. The basic form of getting to know about individuals in a population is to get responses for common benchmarked questions and rate them depending on their responses. This research is a pilot work for further in depth study about different kinds of mental illness. The target population 1 was decided to be high school/college going students and the target population 2 was working professionals with less than 5 years

experience from different organisations. The questionnaire for the target group was framed in consultation with a psychologist. As per the expert's suggestion we decided to prepare the questionnaire to achieve the objective of identifying deviations in mental health. The final set of questions prepared was based on the work by Kern, et al. It supports five different factors that together support higher levels of well-being: engagement, perseverance, optimism, connectedness, and happiness. Based on scores obtained by respondents, a person's mental health status is predicted as mentally distressed, barely satisfied with life and optimistic.

6. Paper Title: Prediction of Mental Health Problems among Children Using Machine Learning Techniques.

Authors name: Ms. Sumathi M.R., Research Scholar, and Dr. B. Poorna

Ms. Sumathi M.R., Research Scholar, and Dr. B. Poorna [6] they proposed about the Prediction of Mental Health Problems Among Children Using Machine Learning Techniques. Machine learning Techniques are currently well suited for analyzing medical data and diagnosing the problem. This research has identified eight machine learning techniques and has compared their performances on different measures of accuracy in diagnosing five basic mental health problems. A data set consisting of sixty cases is collected for training and testing the performance of the techniques. Twenty-five attributes have been identified as important for diagnosing the problem from the documents. The attributes have been reduced by applying Feature Selection algorithms over the full attribute data set. The accuracy over the full attribute set and selected attribute set on various machine learning techniques have been compared. It is evident from the results that the three classifiers viz., Multilayer Perceptron, Multiclass Classifier and LAD Tree produced more accurate results and there is only a slight difference between their performances over full attribute set and selected attribute set.

CHAPTER 3

SOFTWARE AND HARDWARE REQUIREMENTS

3.1 Hardware requirements:

- Windows OS with Intel/AMD 64bit processor and above
- 1GB or greater RAM
- 500MB minimum available memory
- Peripherals - ROM, keyboards , pointing devices, network devices, etc.

3.2 Software requirements:

3.2.1 Python:

Python is an interpreted, high-level, general-purpose programming language. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

3.2.2 Jupitar Lab IDE:

Jupyter Lab is a web-based interactive development environment for Jupyter notebooks, code, and data. Jupyter Lab is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning. Jupyter Lab is extensible and modular: write plugins that add new components and integrate with existing ones. To develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.

3.2.3 Java script:

JavaScript often abbreviated as JS, is a programming language that conforms to the ECMA Script specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions. As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working

with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM). However, the language itself does not include any input/output (I/O), such as networking, storage, or graphics facilities, as the host environment (usually a web browser) provides those APIs. JavaScript engines were originally used only in web browsers, but they are now embedded in some servers, usually via Node.js. They are also embedded in a variety of applications created with frameworks such as Electron and Cordova.

3.2.4 HTML:

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

3.2.5 CSS:

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

3.2.6 Visual Studio:

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Visual Studio Code is a source code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js and C++. It is based on the Electron framework, which is used to develop Node.js web apps that run on the Blink layout engine. Visual Studio Code can be extended via extensions, available through a central repository. This includes additions to the editor and language support. Visual Studio Code includes basic support for most common programming languages. This basic support includes syntax highlighting, bracket matching, code folding, and configurable snippets. Visual Studio Code also ships with IntelliSense for JavaScript, TypeScript, JSON, CSS, and HTML, as well as debugging support for Node.js.

3.3 System Requirements:

The user should have an active internet connection and a desktop computer with minimum 500 MB RAM and Intel/AMD processor with windows (xp/10) or linux installed.

3.4 Hardware Platform:

- Intel/AMD Processor or any other compatible Processor, 1GHz or greater
- Minimum 500 B of RAM capacity or more
- Recommended hard disk space of 1 GB or more

3.5 Software Platform:

- Operating system : Windows/Linux/Mac
- Technology Used : Rectjs
- IDE : Jupitar Lab Python
- Framework : Java Script

3.6 External Interface Requirements:

3.6.1 User Interfaces:

- Home Screen, The home screen is the main user interface that helps the user to log on to the web app.
- HTTP protocols, As this is a web based application, the communication takes place through the HTTP protocols

3.6.2 Software Interfaces:

This project includes software component such as

- Visual Studio Code
- Jupitar Lab

3.7Reactjs :

- React (also known as React.js or ReactJS) is a JavaScript library for building user interfaces. It is maintained by Facebook and a community of individual developers and companies.
- React can be used as a base in the development of single-page or mobile applications, as it is optimal for fetching rapidly changing data that needs to be recorded. However, fetching data is only the beginning of what happens on a web page, which is why complex React

applications usually require the use of additional libraries for state management, routing, and interaction with an API; Redux, React Router and axios are respective examples of such libraries.

- React is a library for building composable user interfaces. It encourages the creation of reusable UI components, which present data that changes over time. Lots of people use React as the V in MVC. React abstracts away the DOM from you, offering a simpler programming model and better performance.
- React can also render on the server using Node, and it can power native apps using React Native. React implements one-way reactive data flow, which reduces the boilerplate and is easier to reason about than traditional data binding.

React Features:

- JSX − JSX is JavaScript syntax extension. It isn't necessary to use JSX in React development, but it is recommended.
- Components − React is all about components. You need to think of everything as a component. This will help you maintain the code when working on larger scale projects.
- Unidirectional data flow and Flux − React implements one-way data flow which makes it easy to reason about your app. Flux is a pattern that helps keeping your data unidirectional.
- License − React is licensed under the Facebook Inc. Documentation is licensed under CC BY 4.0.

CHAPTER 4

SYSTEM DEVELOPMENT PROCESS

System development process is the phase that bridges the gap between the problem domain and the existing system in a manageable way. This phase focuses on the solution domain, i.e. "how to implement?" Specification talks about the hardware and software requirements needed to fulfill or run the project. Basically, the minimum requirements on which the program or app will run.

4.1 SYSTEM DESIGN:

It is the phase where the SRS document is converted into a format that can be implemented and decides how the system will operate. In this phase, the complex activity of system development is divided into several smaller sub activities, which coordinates with each other to achieve the main objective of system development process.

4.2 SYSTEM SPECIFICATION:

The System Specification describes the functional and non-functional requirements posed on a system element In order to prepare the System Specification, the requirements will be derived from the specifications of higher system elements or from the Overall System Specification. The specification provides standards and tools for designing and decomposing the architecture. If changes are required in the course of the development of the system element, the System Specification shall be adapted at first.

4.3FEASIBILITY STUDY:

Feasibility study describes and evaluates candidate systems and provides for the selection of the best candidate system that meets the system performance requirements. Three key considerations are involved in the feasibility analysis:

4.3.1 Economic Feasibility

4.3.2 Technical Feasibility

4.3.3 Behavioral Feasibility

4.3.1.Economic Feasibility:

Economic feasibility determines the benefits and savings that are expected from the system and compare them with costs. Cost/Benefits analysis has been done on the basis of total cost of the system and direct and indirect benefits derived from the system. The total cost of the system comprises of hardware costs and software costs. The main of the economic feasibility is to check whether the system is financially affordable or not. The cost for the proposed system can be divided into two parts given below:

- Hardware Costs :The hardware cost for the proposed system can be calculated from cost of hardware need for the development of the proposed system.
- Software Costs: The software costs for the proposed system can be calculated from the cost of software tools needed from the development of the proposed system.

4.3.2.Technical Feasibility:

Technical Feasibility Centre's on the existing system and to what extent it can support the proposed the system. In this part of feasibility analysis we determined the technical possibilities for the implantation of the system. Two major benefits are:

- Improving the performance
- Minimizing the loss of data during the processing

This project is technically feasible because analysis is made using the Jupitar IDE. This tool has a greater performance when compared to the other analytics tool that's available in the market.

4.3.3.Behavioral Feasibility:

Behavioral feasibility estimates the accuracy of the data resulted towards the development of models of the computerized system. For the analysis of the use case proposed, the measure taken for the creation of the resultant output while with the limitations of the resources and challenging in finding it.

4.4 METHODOLOGY

As per the research, the project plays a vital role in predicting the mental health of the patients.

4.4.1 Existing System:

As the first step for the research, the problem of diagnosing basic mental health was identified and an Interview was held with a clinical psychologist to identify the mental health problems that occur more often among humans. Then, observation was made on how the diagnoses were performed by the professionals. A model was built that uses machine learning techniques to diagnose five common mental health problems effectively. This model assists the professionals to identify the problem if the known evidences of the patient are given as input.

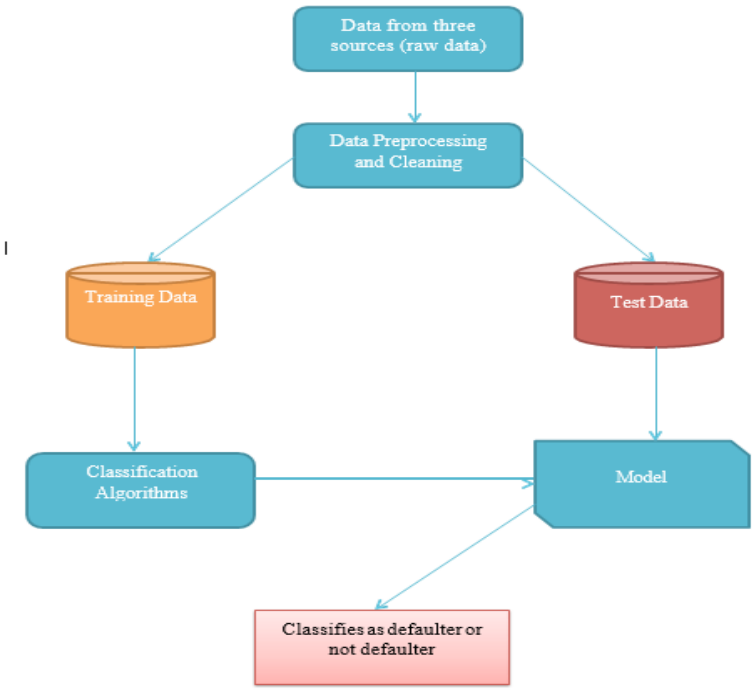

Figure 4.1 Flow Chart for Existing System

Current existing system of the mental health analysis is only up to the defaulter classifiers. At present it is just predicting the mental health issues by taking the common issues in the society. As per the input given in data, system machine predict the problem and gives a default prediction.

4.4.2 Disadvantage of Existing System

- In this current Existing system, we will not get answers in a natural language instead will get it in a machine proposed language.
- At present there is no machine learning based website for the mental health analysis.

4.4.3 Proposed System:

In the proposed system, we are implementing a AI based Chatbot and natural language processing system to process the output for the mental health patients and we are also making a user interface to communicate easily with the machine. Another common barrier to seeking other people's support concerns issues of trust and confidentiality. For example, many young people feel that their problems are too personal to be discussed with anyone or they fear that their sensitive information will be shared with others, and particularly younger adolescents consider it easier to maintain their privacy in online conversations. These individuals may find a fully automated digital intervention that could be used anonymously to be a suitable alternative outlet for disclosure of their mental health difficulties. Similarly, a recent study found that adults were more willing to disclose personal information when they thought they were talking to a nonhuman 'virtual therapist' than a human operating platform.

Finally, if effective, e-Mental Health services may represent the less daunting bridge to getting professional help. It is possible that, as individuals use these resources and learn about cognitive behavioural and other clinical approaches, they might develop skills to recognise when they need additional support and become progressively more open to talking to a real human. For patients who seek face-to-face help first, mental health apps may still be recommended by mental health professionals as a supplement to therapy, or as a form of intermediate support while on the waiting list. There is some evidence that the first generation of mental health digital interventions can be effective for conditions such as anxiety and depression. However, as previously discussed, such interventions are also characterised by poor adherence, which may be due to the lack of the quality of human interaction that a therapist-patient relationship offers. This is particularly the case for unguided interventions, which suggests that complementing digital help with external human support may be a way to increase adherence to e-Mental Health services. However, such option is

Data from counselling

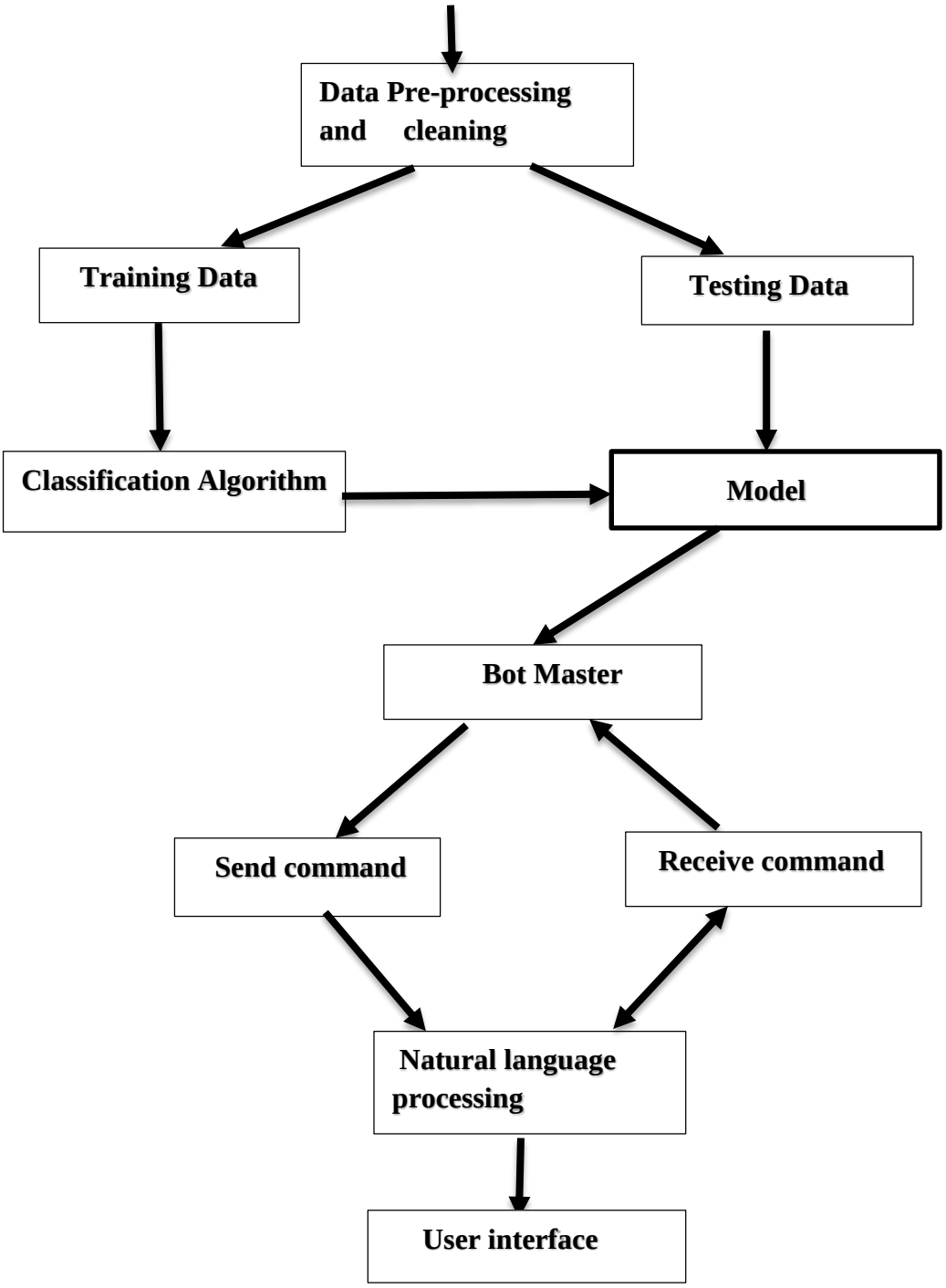

Figure 5.2 Flow Chart for Proposed System

likely to be difficult to implement on a large scale.11 Therefore, fully automated conversational agents, or chatbots, could offer a promising alternative.

Taken together, these initial findings are encouraging, but of course warrant replication with larger and more diverse samples, including clinical populations. It is also important to compare these interventions to face-to-face treatments to ascertain the magnitude of the effect in comparison to traditional methods.

4.4.4 Advantage Of Proposed System:

- Advanced AI chatbots
- Use of Natural Language Processing
- Use of Reactjs user interface

4.5 SYSTEM IMPLEMENTATION

A software application in general is implemented after navigating the complete life cycle method of a project. Various life cycle processes such as requirement analysis, design phase, verification, testing and finally followed by the implementation phase result in a successful project management. System implementation is an important stage of theoretical design is turned into practical system.

4.5.1 Implementation Procedure

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of change over methods.

Each program is tested individually at the time of development using the data and has verified that this program linked together in the way specified in the programs specification, the computer system and its environment is tested to the satisfaction of the user. The system that has been developed is accepted and proved to be satisfactory for the user and so the system is going to be implemented very soon. A simple operating procedure is included so that the user can understand the different functions clearly and quickly. The final stage is to document the entire system which provides components and the operating procedures of the system.

4.5.2 Framework used for Implementation:

a) Reactjs

- React (also known as React.js or ReactJS) is a JavaScript library for building user interfaces. It is maintained by Facebook and a community of individual developers and companies.
- React can be used as a base in the development of single-page or mobile applications, as it is optimal for fetching rapidly changing data that needs to be recorded. However, fetching data is only the beginning of what happens on a web page, which is why complex React applications usually require the use of additional libraries for state management, routing, and interaction with an API; Redux, React Router and axios are respective examples of such libraries.
- React is a library for building composable user interfaces. It encourages the creation of reusable UI components, which present data that changes over time. Lots of people use React as the V in MVC. React abstracts away the DOM from you, offering a simpler programming model and better performance.
- React can also render on the server using Node, and it can power native apps using React Native. React implements one-way reactive data flow, which reduces the boilerplate and is easier to reason about than traditional data binding.

b) Jupitar Note Book:

- Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebook documents.
- The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context.
- A Jupyter Notebook document is a JSON document, following a versioned schema, and containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media, usually ending with the ".ipynb" extension.

- A Jupyter Notebook can be converted to a number of open standard output formats (HTML, presentation slides, LaTeX, PDF, ReStructuredText, Markdown, Python) through "Download As" in the web interface, via the nbconvert library or "jupyter nbconvert" command line interface in a shell.
- To simplify visualisation of Jupyter notebook documents on the web, the nbconvert library is provided as a service through NbViewer which can take a URL to any publicly available notebook document, convert it to HTML on the fly and display it to the user.

4.5.3 Types of Framework used:

There are mainly three types of frameworks:

- **Data Driven Test Framework:** In a data driven framework all of our test data is generated from some external files like Excel , CSV, XML, or some database table.
- **Keyword Driven Test Framework** : In a keyword-driven test framework, all the operations and instructions are written in some external file like Excel worksheet. According to the keywords written in Excel file, the framework will perform the operation on UI.
- **Hybrid Test Framework** : Hybrid Test framework is a concept where one can take advantage of both keyword and data driven framework.

4.5.4 Implementation Module:

1. Home Page Module:

The home page module is user friendly. It holds a useful Covid-19, effective videos and importantly a Chatbot called INSPIRED. In this module a contact page has been created in that page contains fields of name, email id and a message which user whant to share with us. It also maintains as a registration.

2. Inspired Chatbot:

A chatbot is a bot type that simulates human conversation (a chat). Chatbots interact with humans via a (live) chat interface or SMS. Live chat presence on the website is indicated by the chat widget. The more input and feedback they receive the better experience in chatting they build up. The process is similar to human reasoning.

3. Covid-19 Dashboard:

Our website also contains Covid-19 live updates. It's a global wise live updates. It contains Infected, recovered and death rates which is also represented in charts.

4. Inspiring Videos:

This also contains an inspiring videos.

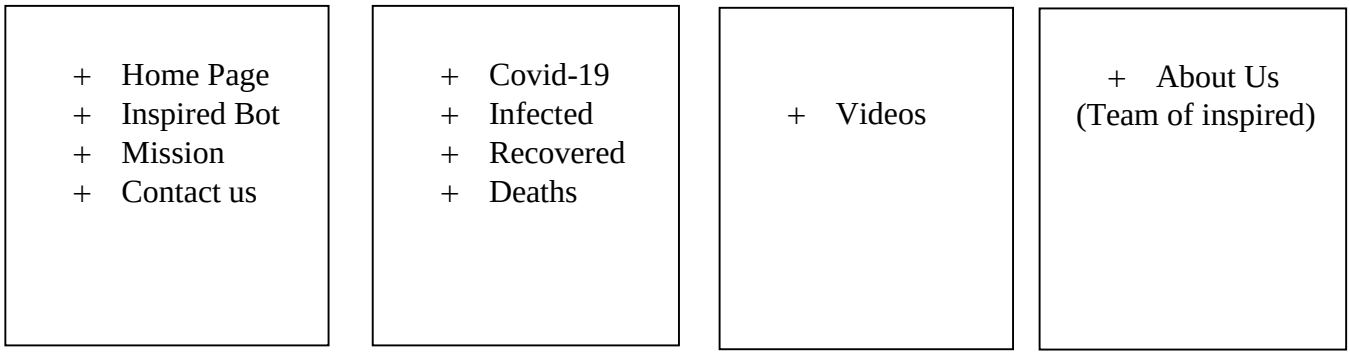

Fig 4.3:
Structure of Page Object

4.6 TESTING

Software testing is nothing but an art of investigating software to ensure that its quality under test is in line with the requirement of the client. Software testing is carried out in a systematic manner with the intent of finding defects in a system. It is required for evaluating the system. Testing is important because software bugs could be expensive or even dangerous. Software bugs can potentially cause monetary and human loss, and history is full of such examples.

4.6.1 Properties of Tests:

1. Fast: Tests give us confidence that our code is working as intended. A slower feedback loop hampers development as it takes us longer to find out if our change was correct. If our workflow is plagued by slow tests, we won't be running them as often. This will lead to problems down the line.

2. Deterministic: Tests should be deterministic, i.e. the same input will always result in the same output. If tests are non-deterministic, we have to find a way to account for random behaviour inside of our tests. While there is definitely non-deterministic code in production (i.e. Machine Learning

and AI), we should try to make all our non-probabilistic code as deterministic as possible. There is no point of doing additional work unless our program requires it.

3. Automated: We can confirm our program works by running it. This could be manually running a command in the or refreshing a webpage; in both cases, we are looking to see if our program does what it is supposed to do. While manual testing is fine for small projects, it becomes unmanageable as our project grows in complexity. By automating our test suite, we can quickly verify our program works on-demand. Some developers even have their tests triggered to run on file save.

4.6.2 Testing Types:

1. Functional Testing:

It's a type of testing where by the system is tested against the functional requirements/specifications. Functions (or features) are tested by feeding them input and examining the output. Functional testing ensures that the requirements are properly satisfied by the application. This type of testing is not concerned with how processing occurs, but rather, with the results of processing. It simulates actual system usage but does not make any system structure assumptions. During functional testing, Black Box Testing technique is used in which the internal logic of the system being tested is not known to the tester. Functional testing is normally performed during the levels of System Testing and Acceptance Testing.

Typically, functional testing involves the following steps:

- Identify functions that the software is expected to perform.
- Create input data based on the function's specifications.
- Determine the output based on the function's specifications.
- Execute the test case.
- Compare the actual and expected outputs.

Functional testing is more effective when the test conditions are created directly from user/business requirements. When test conditions are created from the system documentation (system requirements/ design documents), the defects in that documentation will not be detected through testing and this may be the cause of end-users' wrath when they finally use the software.

2. Integration Testing:

Integration testing is the phase in testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements. It occurs after unit testing and before validation testing.

3. Regression Testing:

Regression testing is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features. Regression Testing is nothing but a full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine.

4.6.3 Coding:

#JavaScript for Covid-19 tracker:

```
import React, { Component } from 'react';
import Nav from "./nav";
class covid extends Component {
  render() {
    return (
      <header>
      <Nav/>
        <div className="head">
          <h1>Covid Tracker</h1>
          <div>
            <p classname="test">Covid Tracker </p>
            <div><a className="contact" href="#">Click to View</a></div>
          </div>
        </div>
      </header>
    );
  }
}
export default covid;
```

#to get covid-19 cards:

```
import React from 'react';
import { Card, CardContent, Typography, Grid } from '@material-ui/core';
import CountUp from 'react-countup';
import cx from 'classnames';
import styles from './Cards.module.css';
const Info = ({ data: { confirmed, recovered, deaths, lastUpdate } }) => {
  if (!confirmed) {
    return 'Loading...';
```

```
}
return (
  <div className={styles.container}>
    <Grid container spacing={3} justify="center">
      <Grid item xs={12} md={3} component={Card} className={cx(styles.card, styles.infected)}>
        <CardContent>
          <Typography color="textSecondary" gutterBottom>
            Infected
          </Typography>
          <Typography variant="h5" component="h2">
            <CountUp start={0} end={confirmed.value} duration={2.75} separator="," />
          </Typography>
          <Typography color="textSecondary">
            {new Date(lastUpdate).toDateString()}
          </Typography>
          <Typography variant="body2" component="p">
            Number of active cases of COVID-19.
          </Typography>
        </CardContent>
      </Grid>
      <Grid item xs={12} md={3} component={Card} className={cx(styles.card, styles.recovered)}>
        <CardContent>
          <Typography color="textSecondary" gutterBottom>
            Recovered
          </Typography>
          <Typography variant="h5" component="h2">
            <CountUp start={0} end={recovered.value} duration={2.75} separator="," />
          </Typography>
          <Typography color="textSecondary">
            {new Date(lastUpdate).toDateString()}
          </Typography>
          <Typography variant="body2" component="p">
            Number of recoveries from COVID-19.
          </Typography>
        </CardContent>
      </Grid>
      <Grid item xs={12} md={3} component={Card} className={cx(styles.card, styles.deaths)}>
        <CardContent>
          <Typography color="textSecondary" gutterBottom>
            Deaths
          </Typography>
          <Typography variant="h5" component="h2">
            <CountUp start={0} end={deaths.value} duration={2.75} separator="," />
          </Typography>
          <Typography color="textSecondary">
            {new Date(lastUpdate).toDateString()}
          </Typography>
          <Typography variant="body2" component="p">
            Number of deaths caused by COVID-19.
          </Typography>
        </CardContent>
```

```
      </Grid>
    </Grid>
  </div>
 );
};
export default Info;
```

#to get Covid-19 charts

```
import React, { useState, useEffect } from 'react';
import { Line, Bar } from 'react-chartjs-2';
import { fetchDailyData } from '../../api';
import styles from './Chart.module.css';
const Chart = ({ data: { confirmed, recovered, deaths }, country }) => {
 const [dailyData, setDailyData] = useState({});
 useEffect(() => {
  const fetchMyAPI = async () => {
   const initialDailyData = await fetchDailyData();

   setDailyData(initialDailyData);
  };

  fetchMyAPI();
 }, []);

 const barChart = (
  confirmed ? (
   <Bar
    data={{
     labels: ['Infected', 'Recovered', 'Deaths'],
     datasets: [
      {
       label: 'People',
       backgroundColor: ['rgba(0, 0, 255, 0.5)', 'rgba(0, 255, 0, 0.5)', 'rgba(255, 0, 0, 0.5)'],
       data: [confirmed.value, recovered.value, deaths.value],
      },
     ],
    }}
    options={{
     legend: { display: false },
     title: { display: true, text: `Current state in ${country}` },
    }}
   />
  ) : null
 );
 const lineChart = (
  dailyData[0] ? (
   <Line
    data={{
     labels: dailyData.map(({ date }) => date),
     datasets: [{
```

```
          data: dailyData.map((data) => data.confirmed),
          label: 'Infected',
          borderColor: '#3333ff',
          fill: true,
        }, {
          data: dailyData.map((data) => data.deaths),
          label: 'Deaths',
          borderColor: 'red',
          backgroundColor: 'rgba(255, 0, 0, 0.5)',
          fill: true,
        },
        ],
      }}
    />
  ) : null
 );
 return (
  <div className={styles.container}>
   {country ? barChart : lineChart}
  </div>
 );
};
export default Chart;
```

#Covid-19 country picker

```
import React, { useState, useEffect } from 'react';
import { NativeSelect, FormControl } from '@material-ui/core';
import { fetchCountries } from '../../api';
import styles from './CountryPicker.module.css';
const Countries = ({ handleCountryChange }) => {
 const [countries, setCountries] = useState([]);

 useEffect(() => {
  const fetchAPI = async () => {
   setCountries(await fetchCountries());
  };
  fetchAPI();
 }, []);
 return (
  <FormControl className={styles.formControl}>
   <NativeSelect defaultValue="" onChange={(e) => handleCountryChange(e.target.value)}>
    <option value="">Global</option>
    {countries.map((country, i) => <option key={i} value={country}>{country}</option>)}
   </NativeSelect>
  </FormControl>
 );
};
export default Countries;
```

#to create Nav page:(.js)

```
import React, { Component } from 'react';
import { Link } from "react-router-dom";
class Nav extends Component {
  render() {
    return (
      <nav>
        <ul>
          <li className="logo">Inspired Chatbot</li>
        </ul>
        <ul>
          <li><a href="/Home">Home</a></li>
          <li><Link to="/corona">Covid-19</Link></li>
          <li><a href="/Videos">Videos</a></li>
          <li><a href="/Aboutus">About Us</a></li>
        </ul>
      </nav>
    );
  }
}
export default Nav;
```

#to create Main Page: (.js)

```
import React, { Component } from 'react';
import Form from "./form";
class Main extends Component {
  render() {
    return (
      <main>

        <section className="intro">
          <h2>About Us</h2>
          <div>
           <p>We as Team Smash want to impact everyone who visits our website. We want to do the least desired job of all to listen to your problems have empathy with the situations you go through .</p>
          </div>
        </section>

        <div>
          <div className="services">
            <div className="service-one">
            <p className="service-icon"><i className="fa fa-share-alt"></i></p>
            <p className="service-title">Sharing</p>
            <p>Share your problems with Smash everything remains a secret the data you share is not stored so no worries.</p>
            </div>
            <div className="service-two">
            <p className="service-icon"><i className="fa fa-heart"></i></p>
            <p className="service-title">Loving</p>
```

```
            <p>We as the team of Smash just desire one thing to spread love because everyone deserves
it.</p>

            </div>
            <div className="service-three">
            <p className="service-icon"><i className="fa fa-globe"></i></p>
            <p className="service-title">Caring</p>
            <p>We know that everyone goes through problems. We care and empathise with what you go
through.</p>

          </div>
        </div>
          </div>
        <div className="gallery">
         <div className="gallery-item-one"></div>
         <div className="gallery-item-two"></div>
         <div className="gallery-item-three"></div>
         <div className="gallery-item-four"></div>
         <div className="gallery-item-five"></div>
         <div className="gallery-item-six"></div>
        </div>
        <section>
          <h2>Our Mission</h2>
          <div>
            <p>To make the world a better place where people are felt important and valued for what they
are.</p>
            <p>We dont Jurdge we just empathise and share love which is in abundance.</p>
            </div>
        </section>
        <section>
          <h2>Contact Us</h2>
          <Form />
        </section>
      </main>
    );
  }
}
export default Main;
```

#header in main page:(.js)

```
import React, { Component } from 'react';
import Nav from "./nav";
class Header extends Component {
  render() {
    return (
      <header>
      <Nav/>
```

```
        <div className="head">
          <h1>Inspired Chatbot</h1>
          <div>
            <p className="test">Talk to Inspired chatbot and experience the human touch of a robot, Share
your emotions with smash the councelling expert. </p>
            <div><a className="contact" href="#">Contact Us</a></div>
          </div>
        </div>

      </header>
    );
  }
}
export default Header;
```

#to create contact info in main page: (.js)

```
import React, { Component } from 'react';
class Form extends Component {
  constructor(){
     super();
     this.state = {
        name : "",
        email : "",
        message : "",
        formError: false
     }
  }
  getName = (e) =>{
    let username = e.target.value;
    this.setState({
       name: username
    });
    console.log(this.state.name);
  }
  getEmail = (e) =>{
    let userEmail = e.target.value;
    //the most important thing is that we use a RegEx
    //in order to manage the input of the email
    //at least we can get a some what valid email
    if(userEmail.match(/^([a-zA-Z0-9_\-\.]+)@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.)|(([a-zA-Z0-9\-
]+\.)+))([a-zA-Z]{2,4}|[0-9]{1,3})(\]?)$/)){
      this.setState({
        email: userEmail
    });
      }else{
        this.setState({
          email: ""
      });
      console.log("Incorrect Email, must match Expression");
```

```
    }

    console.log(this.state.email);
  }
  getDescription = (e) =>{
    let userMessage = e.target.value;
    this.setState({
       message: userMessage
    });
    console.log(this.state.message);
  }
  //send the form
  submitForm = (e) =>{
    e.preventDefault();
    if(this.state.name === "" || this.state.email === "" || this.state.message === "" ){
     this.setState({
       formError: true
     })
      return false;
    }else{
       this.setState({
          formError: false
        })
       console.log(`UserData: {
          Username: ${this.state.name},
          Email: ${this.state.email},
          Message: ${this.state.message}
       }`)

       console.log("form sent")
    }
  }
  render() {

    return (
       <form>
         {/* I am just sending a basic error message */}
            {this.state.formError &&
              <p className="error">
                 Fill all the input fields please.
              </p>
            }
            <p>Fill in the next form to send us a message</p>
            <div>
              <label htmlFor="name">Name</label>
              <input type="text" name="name" placeholder="Your name here please"
onChange={this.getName} />
            </div>
            <div>
              <label htmlFor="email">Email</label>
```

```
            <input type="email" name="email" placeholder="We will contact you after reviewing your
message" onChange={this.getEmail} />
          </div>
          <div>
            <label htmlFor="name">Message</label>
            <textarea onChange={this.getDescription} maxLength="450"></textarea>
          </div>
          <div>
            <p>We will answer as soon as possible</p>
            <input type="submit" name="submit" value="Send" onClick= {this.submitForm} />
          </div>
        </form>
    );
  }
}
export default Form;
```

#to create contact info (social media) in main page: (.js)

```
import React, { Component } from 'react';
class Footer extends Component {
  render() {
    return (
      <footer>
          <h3>Made in India With ❤</h3>
          <p>Support <br/> cmr.edu.in</p>
          <ul>
            <li><a href="#"><i className="fab fa-facebook-f"></i></a></li>
            <li><a href="#"><i className="fab fa-twitter"></i></a></li>
            <li><a href="#"><i className="fab fa-instagram"></i></a></li>
          </ul>
      </footer>
    );
  }
}
export default Footer;
```

#Code For ChatBot:

```
import nltk

from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

import pickle
```

```
import numpy as np
from keras.models import load_model
model = load_model('chatbot_model.h5')
import json
import random
intents = json.loads(open('intents.json').read())
words = pickle.load(open('words.pkl','rb'))
classes = pickle.load(open('classes.pkl','rb'))
def clean_up_sentence(sentence):
sentence_words = nltk.word_tokenize(sentence)
sentence_words = [lemmatizer.lemmatize(word.lower()) for word in sentence_words]
return sentence_words
def bag_of_words(sentence, words, show_details=True):
sentence_words = clean_up_sentence(sentence)

# bag of words — vocabulary matrix
bag = [0]*len(words)
for s in sentence_words:
for i,word in enumerate(words):
if word == s:

# assign 1 if current word is in the vocabulary position
bag[i] = 1
if show_details:
print ("found in bag: %s" % word)
return(np.array(bag))

# filter below threshold predictions
p = bag_of_words(sentence, words,show_details=False)
res = model.predict(np.array([p]))[0]
```

```
results = [[i,r] for i,r in enumerate(res) if r>ERROR_THRESHOLD]

# sorting strength probability
return_list = []
for r in results:
return_list.append({“intent”: classes[r[0]], “probability”: str(r[1])})
return return_list
def getResponse(ints, intents_json):
tag = ints[0][‘intent’]
list_of_intents = intents_json[‘intents’]
for i in list_of_intents:
if(i[‘tag’]== tag):
result = random.choice(i[‘responses’])
return result

#Creating tkinter GUI
import tkinter
def send():
msg = EntryBox.get(“1.0”,’end-1c’).strip()
EntryBox.delete(“0.0”,END)
if msg != ‘’:
ChatBox.config(state=NORMAL)
ChatBox.insert(END, “You: “ + msg + ‘\n\n’)
ChatBox.config(foreground=”#446665", font=(“Verdana”, 12 ))
ints = predict_class(msg)
res = getResponse(ints, intents)
ChatBox.insert(END, “Bot: “ + res + ‘\n\n’)
ChatBox.config(state=DISABLED)
ChatBox.yview(END)
root.title(“Chatbot”)
```

```
root.geometry(“400x500”)
root.resizable(width=FALSE, height=FALSE)

#Create Chat window
ChatBox = Text(root, bd=0, bg=”white”, height=”8", width=”50", font=”Arial”,)
ChatBox.config(state=DISABLED)

#Bind scrollbar to Chat window
scrollbar = Scrollbar(root, command=ChatBox.yview, cursor=”heart”)
ChatBox[‘yscrollcommand’] = scrollbar.set

#Create Button to send message
SendButton = Button(root, font=(“Verdana”,12,’bold’), text=”Send”, width=”12", height=5,
bd=0, bg=”#f9a602", activebackground=”#3c9d9b”,fg=’#000000',
command= send )
#Create the box to enter message
EntryBox = Text(root, bd=0, bg=”white”,width=”29", height=”5", font=”Arial”)
#EntryBox.bind(“<Return>”, send)
scrollbar.place(x=376,y=6, height=386)
ChatBox.place(x=6,y=6, height=386, width=370)
EntryBox.place(x=128, y=401, height=90, width=265)
root.mainloop()
```

CHAPTER 5

RESULTS AND CONCLUSION

The project Mental Health Analysis ChatBot INSPIRED which is built using Reactjs, Python and other tools. The main objective is, Talk to Inspired chatbot and experience the human touch of a robot, Share your emotions with smash the councelling expert.

5.1 RESULTS AND ANALYSIS:

- As we mentioned in the objective of the project we implemented a required contents for the project. The data suggested in the Chatbot the digital mental health interventions are likely to affect young people in a very important dimension of their lives.
- And, the users receive reports that might help them gain insight into their own patterns, as well as targeted therapy exercises, including reframing one's thoughts, mindful breathing, and motivational interviewing, in the form of text, games, or video clips.
- As we know about the current pandemic that is corona, Our team has been successful in making the Covid-19 live dashboard. Because it is also helps keeps people updated and during this pandemic people might also suffer mental stress.
- And we successful in implementing some inspiring videos which may help to overcome mental stress, anxiety etc.,.
- The main mission of the project is, To make the world a better place where people are felt important and valued for what they are. We don't Judge we just empathise and share love which is in abundance.
- We as Team Smash want to impact everyone who visits our website. We want to do the least desired job of all to listen to your problems have empathy with the situations you go through.
- For their potential not to be compromised, we believe that automated bots should meet a set of minimum ethical standards concerning privacy and confidentiality, efficacy, and safety.

5.2 SCREENSHOTS

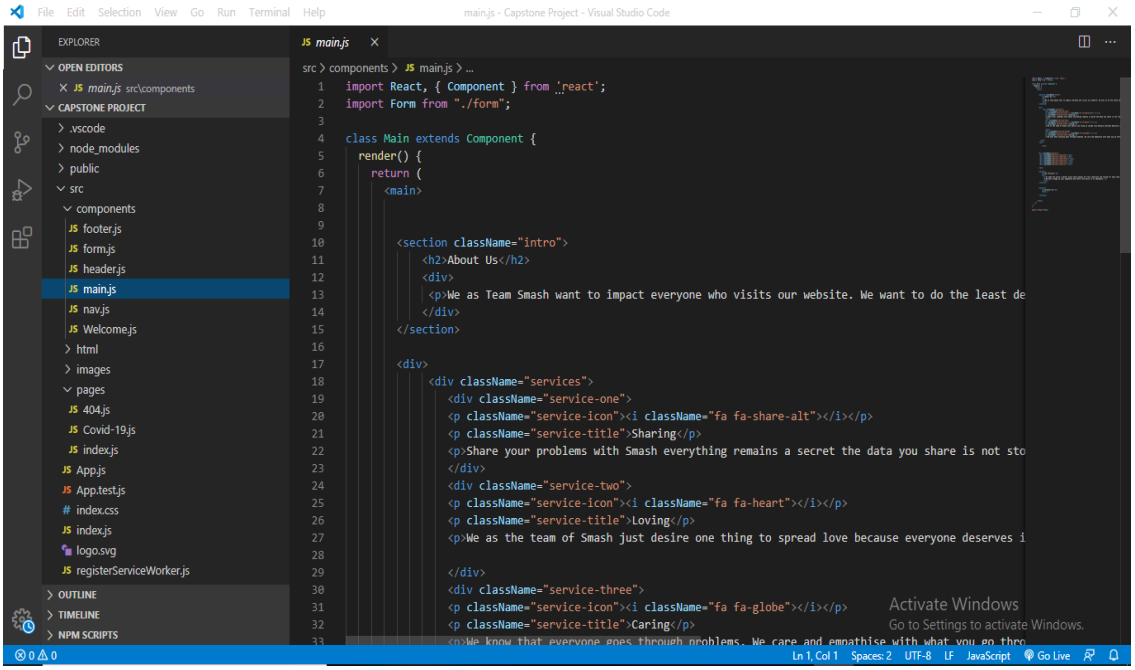

Fig 5.1: Creation Script

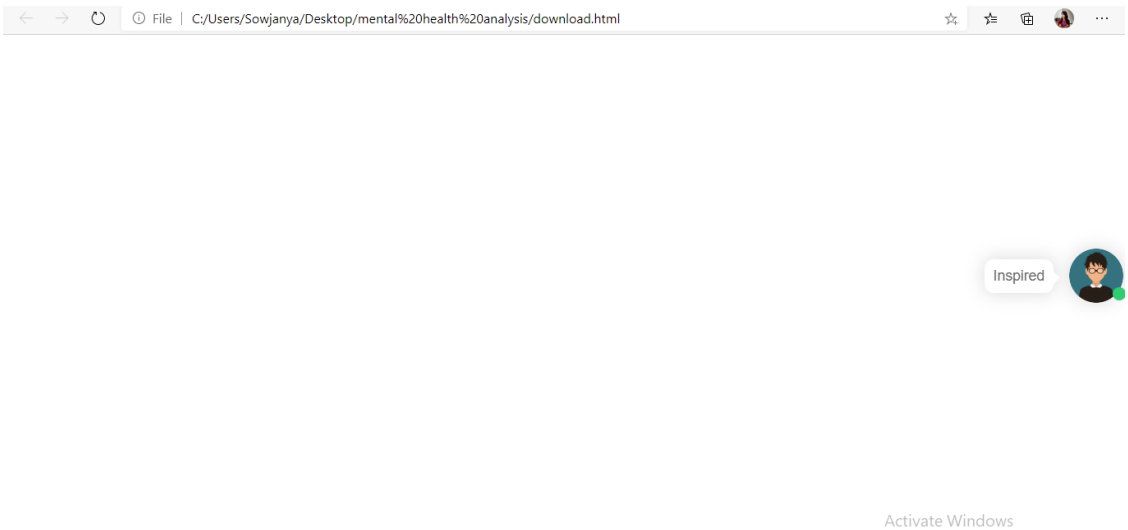

Fig 5.2: Opening browser window

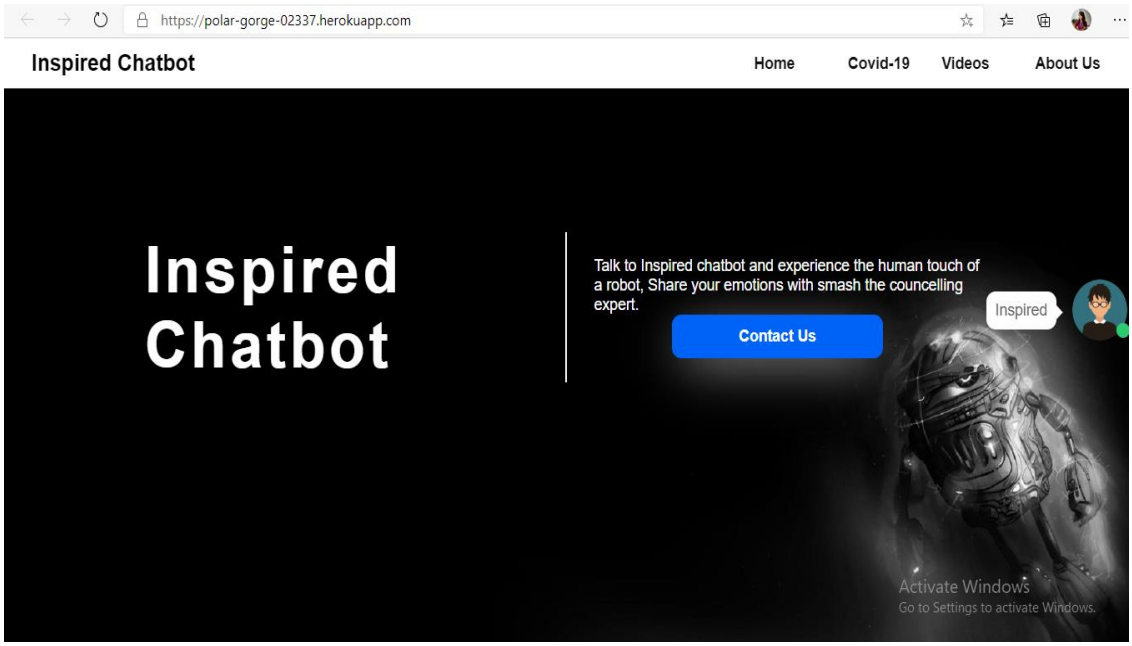

Fig 5.3: Home Page 1

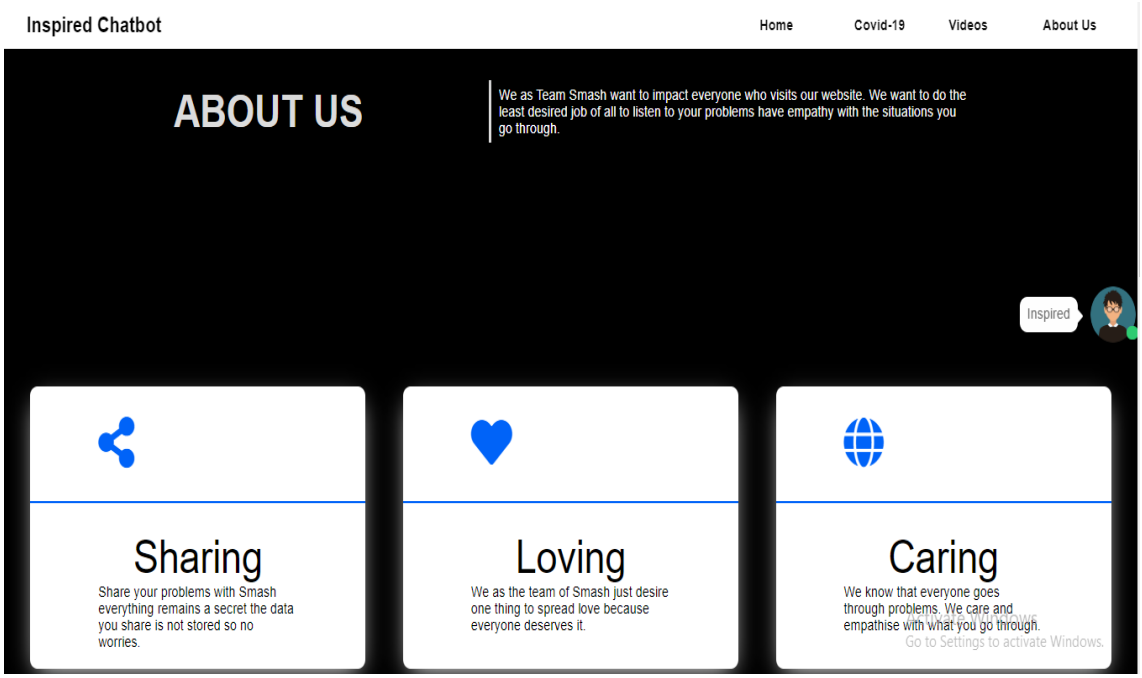

Fig 5.4: Home Page 2

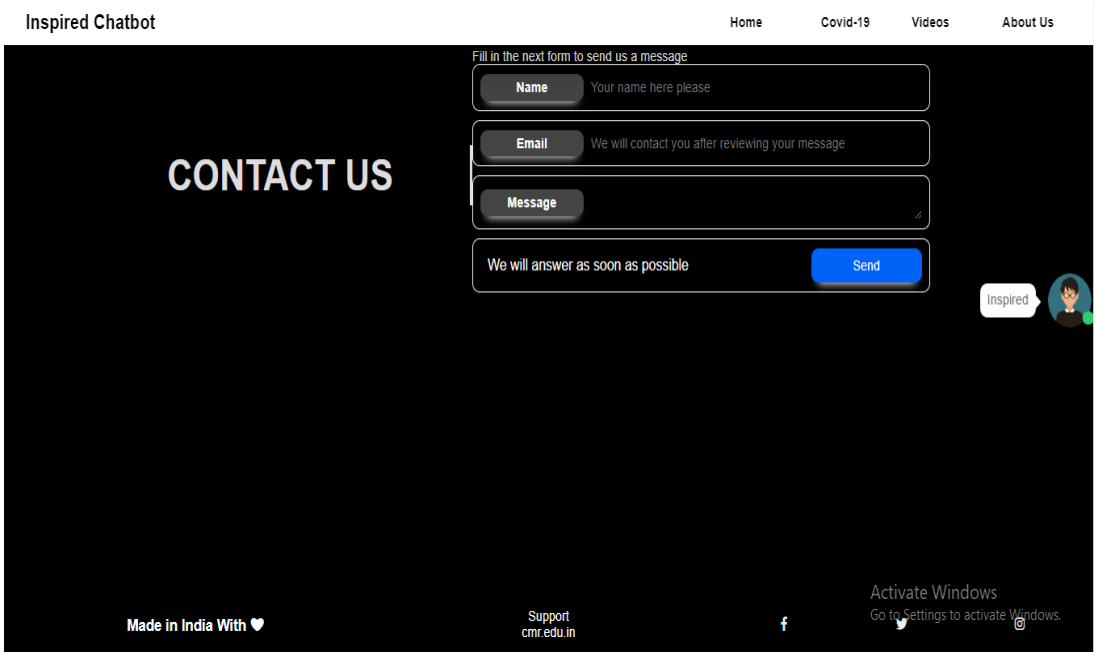

Fig 5.5: Contact Info

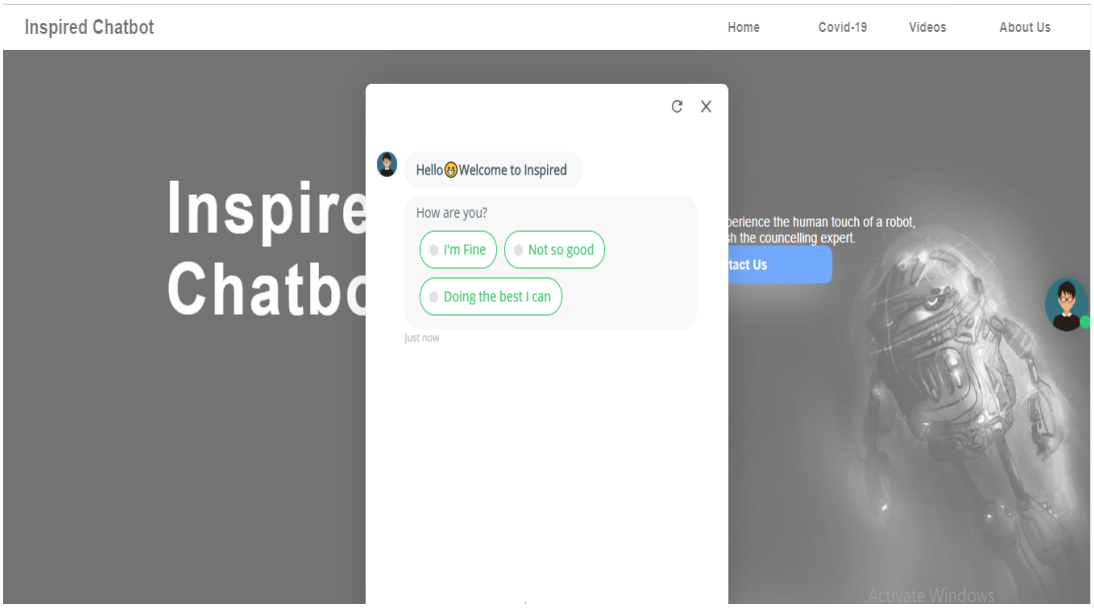

Fig 5.6: ChatBot

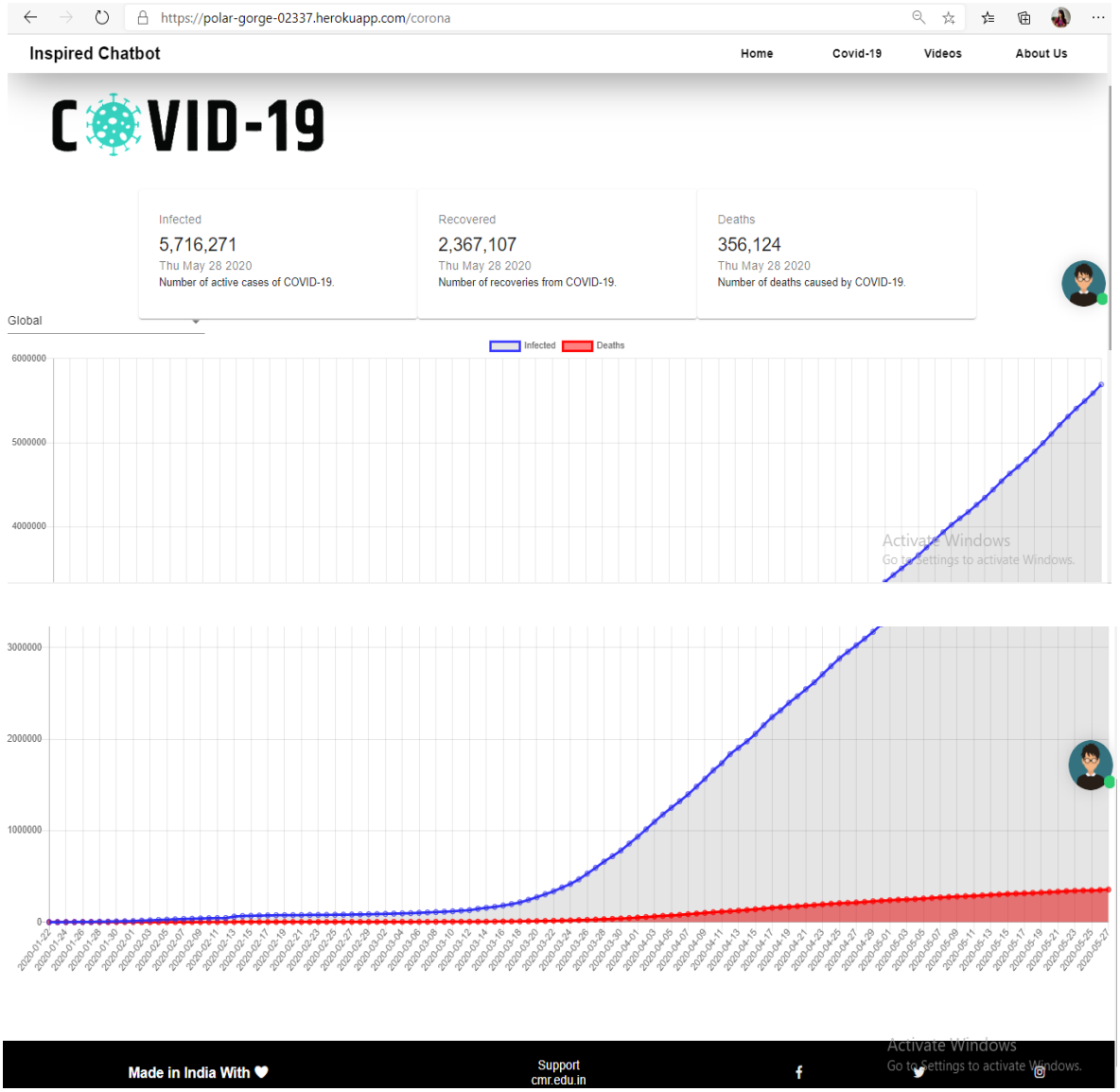

Fig 5.7: Covid-19 Death and Infected Rate in Global

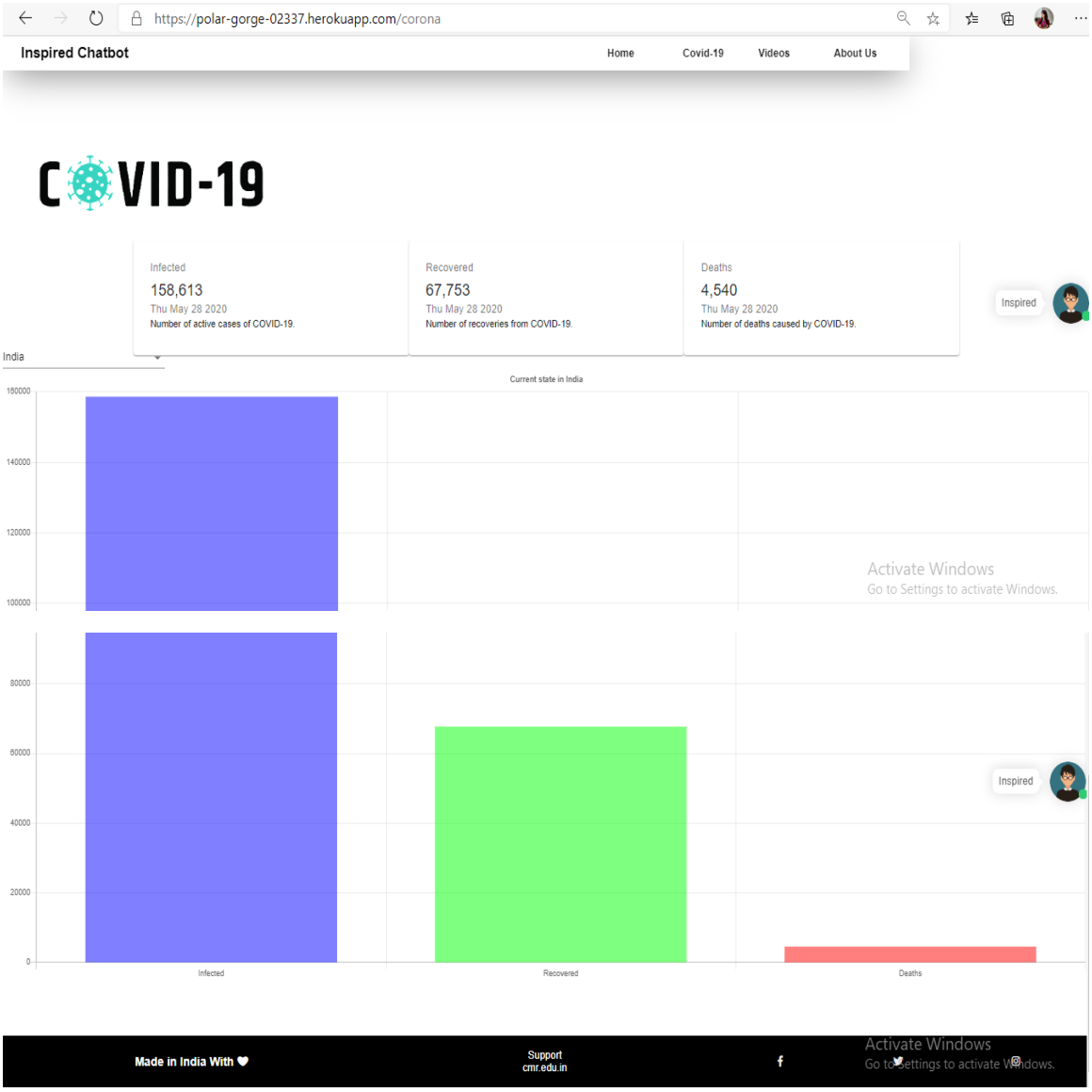

Fig 5.8: Infected, Recovered and Death rate in India

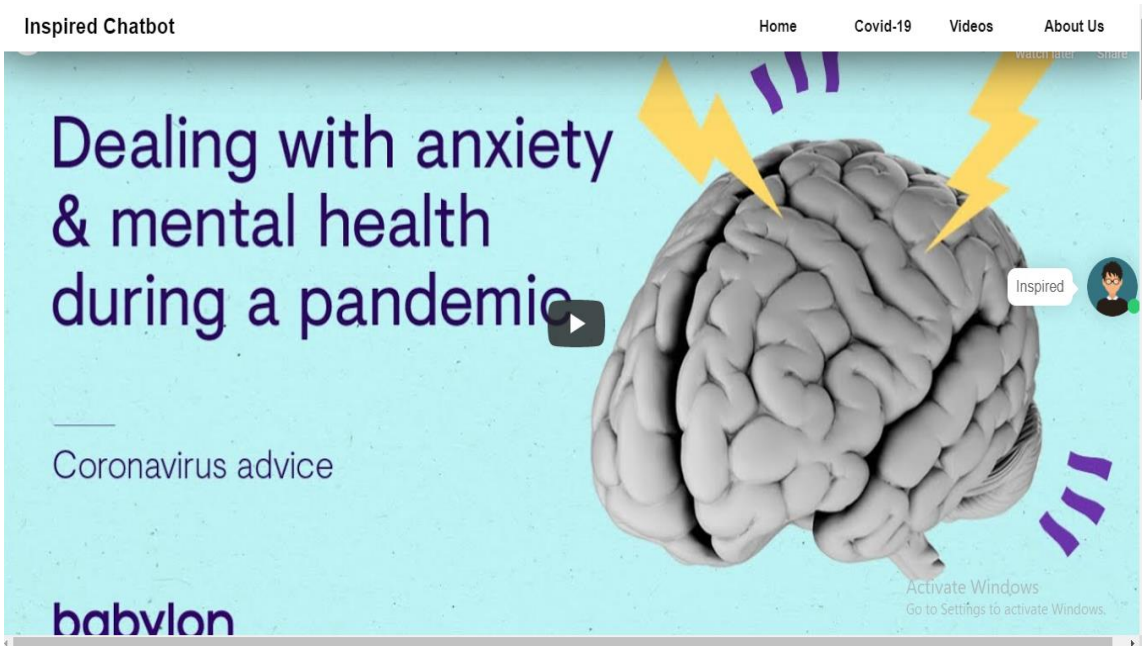

Fig 5.9 Videos Page

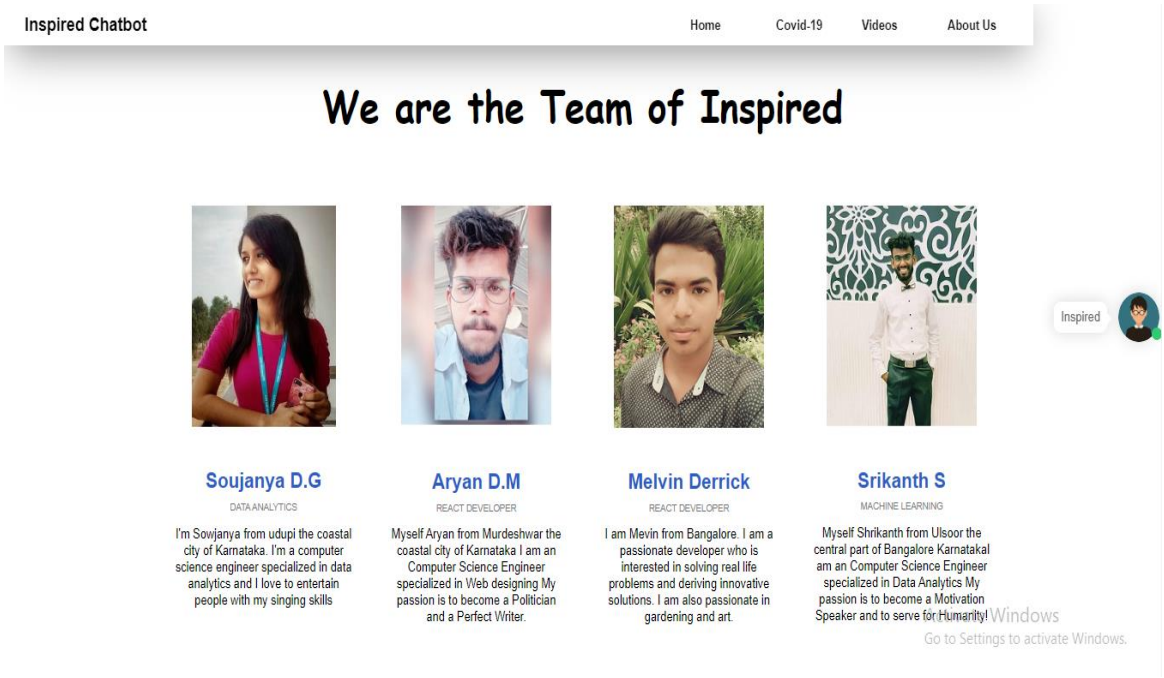

Fig 5.10 About us Page

CONCLUSION

In a development, we might not get enough time to specialize our new features of ChatBot in a time, so we duplicated a lot of code in many places. Refactoring code is an inherent part of software development to avoid building up a huge tech debt. Obviously, continuous improvement is essential if we are going to get a good return on investment from the website.

- We have proposed a new framework React Java Script for designing Web Page.
- The Web page is user friendly and easy to handle.
- As we mentioned in the objective of the project we implemented a required contents for the project. The data suggested in the Chatbot the digital mental health interventions are likely to affect young people in a very important dimension of their lives.
- As we know about the current pandemic that is corona, our team has been successful in making the Covid-19 live dashboard. Because it is also helps keeps people updated and during this pandemic people might also suffer mental stress.
- Successful in implementing some inspiring videos which may help to overcome mental stress, anxiety etc.,

REFERENCES

[1]. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4773121 [Kessler D, Lewis G, Kaur S, et al. Therapist-delivered Internet psychotherapy for depression in primary care: a randomised controlled trial. *Lancet*. 2009;374:628–634. Doi:10.1016/S0140-6736(09)61257-5.]

[2]. https://www.ncbi.nlm.nih.gov/pubmed/24844847 [Arnberg FK, Linton SJ, Hultcrantz M, Heintz E, Jonsson U. Internet-delivered psychological treatments for mood and anxiety disorders : a systematic review of their efficacy, safety, and cost-effectiveness. *PLoS ONE*. 2014;9:e98118. doi:10.1371/journal.pone.0098118.]

[3]. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4820657 [Baumel A, Schueller SM. Adjusting an available online peer support platform in a program to supplement the treatment of perinatal depression and anxiety. *JMIR Ment Health*. 2016;3:e11. doi:10.2196/mental.5335].

[4]. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6402067 [Leykin Y, Muñoz RF, Contreras O, Latham MD. Results from a trial of an unsupported internet intervention for depressive symptoms. *Internet Interv*. 2014;1:175–181. doi:10.1016/j.invent.2014.09.002.Results].

[5]. https://www.ncbi.nlm.nih.gov/pubmed/25093485 [Adelman CB, Panza KE, Bartley CA, Bontempo A, Bloch MH. A metaanalysis of computerized cognitive-behavioral therapy for the treatment of DSM-5 anxiety disorders. *J Clin Psychiatry*.2014;75:e695–e704. doi:10.4088/JCP.13r08894.]

[6]. https://www.tandfonline.com/doi/full/10.1080/16506070903318960 [Andersson G, Cuijpers P. Internet-based and other computerized psychological treatments for adult depression: a meta-analysis. *Cogn Behav Ther*. 2009;38:196–205. doi:10.1080/16506070903318960].

[7]. https://www.ncbi.nlm.nih.gov/pubmed/28546138 [Grist R, Porter J, Stallard P. Mental health mobile apps for preadolescents and adolescents: a systematic review. *J Med Internet Res*. 2017;19:e176. doi:10.2196/jmir.7332].

[8]. https://europepmc.org/articles/PMC2762797 [Christensen H, Griffiths KM, Farrer L. Adherence in Internet interventions for anxiety and depression. *J Med Internet Res*. 2009;11:e13. doi:10.2196/jmir.1194].

[9]. https://www.thelancet.com/journals/lanpsy/article/PIIS2215-0366(18)30230-X/fulltext [Sachan D. Self-help robots drive blues away. *Lancet Psychiatry*. 2018;5:547. doi:10.1016/S2215-0366(18)30230-X].