DETECTION OF KNEE OSTEOATHRITIS AND IT’S SEVERITY USING DEEP LEARNING

##### A PROJECT REPORT

###### ***Submitted by***

##### MUTHU ANNAMALAI.V [211419104171]

##### KOULURU NANDA KISHORE REDDY [211419104142]

##### MUKESH KUMAR.S [211419104169]

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

*in*

# COMPUTER SCIENCE AND ENGINEERING



##### 

PANIMALAR ENGINEERING COLLEGE

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

##### APRIL 2023

**PANIMALAR ENGINEERING COLLEGE**

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

**BONAFIDE CERTIFICATE**

Certified that this project report “DETECTION OF KNEE OSTEOATHRITIS AND IT’S SEVERITY USING DEEP LEARNING” is the bonafide work of MUTHU ANNAMALAI.V [REGISTER NO. 211419104171], KOULURU NANDA KISHORE REDDY [REGISTER NO. 211419104142], MUKESH KUMAR.S [REGISTER NO. 211419104169] who carried out the project work under my supervision.

**SIGNATURE SIGNATURE**

**Dr. L. JABASHEELA, M.E., Ph.D., Dr. G. SENTHIL KUMAR, M.E., Ph.D., PROFESSOR PROFESSOR**

**HEAD OF THE DEPARTMENT SUPERVISOR**

DEPARTMENT OF CSE, DEPARTMENT OF CSE,

PANIMALAR ENGINEERING COLLEGE, PANIMALAR ENGINEERING COLLEGE, NASARATHPETTAI, NASARATHPETTAI,

POONAMALLEE, POONAMALLEE,

CHENNAI - 600 123. CHENNAI - 600 123.

Certified that the above mentioned candidates were examined by us in the End Semester Project Viva-Voice Examination held on at Panimalar Engineering College, Chennai - 600 123.

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**DECLARATION BY THE STUDENTS**

We **MUTHU ANNAMALAI.V** [REGISTER NO. **211419104142**], **KOULURU NANDA KISHORE REDDY** [REGISTER NO. **211419104142**], **MUKESH KUMAR. S** [REGISTER NO. **211419104142**] hereby declare that this project report titled **“DETECTION OF KNEE OSTEOATHRITIS AND IT’S SEVERITY USING DEEP LEARNING”**, under the guidance of **Dr. G. SENTHIL KUMAR, M.C.A., M.Phil., M.B.A., M.E., Ph.D.,** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

**MUTHU ANNAMALAI.V(211419104171)**

**KOULURU NANDA KISHORE REDDY(211419104142)**

**MUKESH KUMAR.S(211419104169)**

**ACKNOWLEDGEMENT**

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr. P. CHINNADURAI, M.A., Ph.D.,** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our beloved Directors **Tmt. C. VIJAYARAJESWARI**, **Dr. C. SAKTHI KUMAR, M.E., Ph.D.,** and **Dr. SARANYA SREE SAKTHI KUMAR B.E., M.B.A., Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr. K. Mani, M.E., Ph.D.,** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. L. JABASHEELA, M.E., Ph.D.,** for the support extended throughout the project.

We would like to thank my **Project Guide**, **Dr. G SENTHIL KUMAR, M.C.A., M.Phil., M.B.A., M.E., Ph.D.,** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

**MUTHU ANNAMALAI.V**

**KOULURU NANDA KISHORE REDDY**

**MUKESH KUMAR.S**

**ABSTRACT**

Osteoarthritis affects 30% of the global population over the age of 60 and is one of the leading causes of disability among older adults. Globally, OA affects more than 250 million people, causing 1–2% of the gross domestic product to be spent on it. Increasing aging in the global population will result in an increase in knee osteoarthritis (KOA) patients. The symptoms of KOA include pain, stiffness, loss of joint range of motion, and gait dysfunctions that worsen with the progression of the disease. It can be detrimental to an individual's quality of life and functional independence to suffer from these symptoms. Using standardized grading systems like Kellgren-Lawrence (KL) grading schemes, experienced radiologists grade the impairment severity [3]. If knee OA is detected and classified early in a patient, corrective measures can be taken and humankind will benefit significantly. Although as of now there is no known cure for arthritis, the benefits of early detection can’t be understated. Our Knee Osteoarthritis Detection project helps patients detect Osteoarthritis in their knees. Along with helping with early detection, this web application also detects the severity of the disorder. According to the severity level of the condition, they can take the appropriate treatment. A severity level is defined by the grades that begin with 0 and go up to 5 (5 grades are indicated) [8].

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** | v |
|  | **LIST OF FIGURES** | ix |
|  | **LIST OF ABBREVIATIONS** | x |
| **1.** | **INTRODUCTION** | 1 |
|  | 1.1 Problem Definition | 2 |
| **2.** | **LITERATURE SURVEY** | 3 |
| **3.** | **SYSTEM ANALYSIS** | 9 |
|  | 3.1 Existing System | 10 |
|  | 3.2 Proposed system | 10 |
|  | 3.3 Feasibility Study | 10 |
|  | 3.3.1 Technical Feasibility | 11 |
|  | 3.3.2 Economic Feasibility | 12 |
|  | 3.3.3 Operational Feasibility | 13 |
|  | 3.4 Hardware Environment | 13 |
|  | 3.5 Software Environment | 14 |
|  | 3.6 Advantages and Disadvantages | 14 |
| **4.** | **SYSTEM DESIGN** | 15 |
|  | 4.1 System Architecture Diagram | 16 |
|  | 4.2 E-R Diagram | 17 |
|  | 4.3 Use Case Diagram | 18 |
|  | 4.4 Class Diagram | 19 |
|  | 4.5 Sequence Diagram | 20 |
|  | 4.5.1 Admin Sequence Diagram | 20 |
|  | 4.5.2 Patient Sequence Diagram | 21 |
|  | 4.6 Activity Diagram | 22 |
|  | 4.6.1 Admin Activity Diagram | 22 |
|  | 4.6.2 Patient Activity Diagram | 23 |
|  | 4.7 Data Flow Diagram | 24 |

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | 4.7.1 Database Detail | 24 |
|  | 4.7.2 Level 1 DFD | 25 |
|  | 4.7.3 Level 2 DFD | 25 |
| **5.** | **SYSTEM IMPLEMENTATION** | 26 |
|  | 5.1 Modules and its Description | 27 |
|  | 5.1.1 Admin Module | 27 |
|  | 5.1.1.1 Login | 27 |
|  | 5.1.1.2 View Patient | 27 |
|  | 5.1.1.3 Logout | 28 |
|  | 5.1.2 Patient Module | 28 |
|  | 5.1.2.1 Login | 28 |
|  | 5.1.2.2 Register | 28 |
|  | 5.2 Project Implementation Technology | 28 |
|  | 5.2.1 Project Lifecycle Details | 29 |
|  | 5.2.1.1 Waterfall Model | 29 |
|  | 5.2.2 Technology Used | 30 |
|  | 5.2.2.1 Python | 30 |
|  | 5.2.2.2 Django | 31 |
|  | 5.2.2.3 XAMPP Server | 33 |
|  | 5.2.2.4 SQL Server | 36 |
| **6.** | **TESTING** | 39 |
|  | 6.1 Levels of Testing | 40 |
|  | 6.1.1 Unit Testing | 40 |
|  | 6.1.2 Integration Testing | 41 |
|  | 6.1.3 System testing | 41 |
|  | 6.1.4 Validation Testing | 42 |
|  | 6.1.5 Output Testing | 42 |
|  | 6.1.6 User Acceptance Testing | 43 |
|  | 6.2 Test Cases | 43 |
|  | 6.2.1 Registration | 43 |
|  | 6.2.2 Login | 43 |

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | 6.2.3 Validation Criteria | 43 |
| **7.** | **CONCLUSION** | 46 |
|  | 7.1 Project Conclusion | 47 |
|  | **APPENDICES** | 48 |
|  | A.1 Project Coding | 49 |
|  | A.2 Snapshots | 79 |
|  | **REFERENCES** | 84 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
|  | **TITLE** | **PAGE NO.** |
| **4.1** | **System Architecture for Admin and Patient** | 24 |
| **4.2** | **E-R Diagram for Whole System** | 25 |
| **4.3** | **Use Case Diagram for Admin and Patient** | 26 |
| **4.4** | **Class Diagram for User and New Registration** | 27 |
| **4.5.1** | **Admin Sequence Diagram** | 28 |
| **4.5.2** | **Patient Sequence Diagram** | 29 |
| **4.6.1** | **Admin Activity Diagram** | 30 |
| **4.6.2** | **Patient Activity Diagram** | 31 |
| **4.7.1** | **Database Detail of User** | 32 |
| **4.7.2** | **Level 1 DFD** | 33 |
| **4.7.3** | **Level 2 DFD** | 33 |
| **5.2.1.1** | **Waterfall Model** | 37 |
| **5.2.2.4** | **SQL Server** | 44 |

|  |  |
| --- | --- |
| **LIST OF ABBREVATIONS** | |
| **OA** | Osteoarthritis |
| **CNN** | Convolutional Neural Networks |
| **CRM** | Customer Relationship Management |
| **SQL** | Structured Query Language |
| **UML** | Unified Modelling Language |
| **DFD** | Data Flow Diagram |

**CHAPTER 1 INTRODUCTION**

**1.INTRODUCTION**

**1.1 PROBLEM DEFINITION**

Arthritis is a general term used to describe any disorder that affects the joints in the body. It is a disease that causes swelling, tenderness, inflammation, stiffness, and a variety of other unpleasant symptoms in a joint or parts of a joint. It is more common in older people than in younger ones, and it generally gets worse with age. Despite the fact that there are a variety of different types of arthritis with a variety of different causes and treatments, osteoarthritis is the most common. Osteoarthritis is the most prevalent form of arthritis in the world and affects almost 237 million people worldwide, or about 3.3% of the total population. Despite the fact that arthritis has no known cure, early detection and treatment can certainly be beneficial in reducing the symptoms of the disease. Our project helps to detect the early signs of osteoarthritis in the knees of patients and to warn them of it as well. Moreover, this project also warns them about the severity and urgency of the disorder that they are suffering from. The most prevalent form of joint disease, osteoarthritis (OA), it impacts 10- 15% of individuals all over the world and can result in long-term pain as well as other problems. There have been 61.2 million cases of OA in China since 1990 [1] . In 2015, according to estimates from the World Health Organization [2], 9.6% of men and 18.0% of women who were 60 years of age or older had symptoms of OA. 25% of those with symptomatic OA are unable to carry out their daily tasks, and 80% of those with mobility limitations. Our Knee Osteoarthritis Detection project helps patients detect Osteoarthritis in their knees by means of the use of the Django framework with Python as the programming language and convolutional neural networks as the algorithm.

**CHAPTER 2** **LITERATURE SURVEY**

**2.LITERATURE SURVEY**

The purpose of a literature review is to provide knowledge gained by reviewing existing research and debates relevant to a particular topic or field of study. Literature reviews contribute to the advancement of knowledge within a specific area of study by contributing to the development of new ideas.

**Title:** Defining the presence of radiographic knee osteoarthritis: a comparison between the Kellgren and Lawrence system and OARSI atlas criteria.

**Description**: For defining knee osteoarthritis (OA) radiographically, Kellgren and Lawrence (K/L) and Osteoarthritis Research Society International (OARSI) atlases are frequently used. This study aimed to assess the extent to which tibiofemoral OA rates differ between the K/L system and the OARSI atlas criteria, and to compare qualitative (K/L and OARSI) and quantitative (milli meters) measures of joint space narrowing (JSN).

**Title:** Quantifying radiographic knee osteoarthritis severity using deep convolutional neural networks

**Description:** The objective of this paper is to develop a new approach that uses convolutional neural networks (CNNs) [14]to automatically estimate knee osteoarthritis (OA) severity from radiographs. A five-point scale is used to assess the severity of knee OA using Kellgren / Lawrence (KL) grades. A variety of hand-engineered features have been used previously to train shallow classifiers to predict KL grades from radiograph images. Using deep convolutional neural networks pre-trained on ImageNet and fine-tuned on knee OA images, the accuracy of classification is significantly improved

**Title:** Automatic knee cartilage segmentation using fully volumetric convolutional neural networks for evaluation of osteoarthritis

**Description:** The authors present an algorithm for segmentation of knee cartilage from a high-resolution magnetic resonance image, using a novel 3D-full convolutional neural network (CNN) called the CNO', along with a multi-class loss function.[15] They claim to have developed the first method for automatically segmenting cartilage using 3D CNNs. MICCAI SKI 10 public challenge results showed that the proposed algorithm performed better than the state-of-the-art algorithm

**Title:** Discovering knee osteoarthritis bone shape features using deep learning

**Description:** Previous studies showed that bone shape and OA are associated, but inferential statistics cannot guarantee the accuracy of actual predictions. A convolutional neural network is used in this study to explore the ability to use knee bone shape features in [16] diagnosing and predicting the onset of radiographic OA based on Kellgren-Lawrence grade in predicting future onset of radiographic OA.

**Title:** Automatic grading of individual knee osteoarthritis features in plain radiographs using deep convolutional neural networks.

**Description:** This study developed an automated method for predicting KL and OARSI grades based on knee radiographs. [13] Using deep learning methods, they leverage residual networks with 50 layers. With the assistance of the Osteoarthritis Initiative (OAI) dataset, they used transfer learning from ImageNet.

**Title:** Learning osteoarthritis imaging biomarkers from bone surface spherical encoding

**Description:** A bone segmentation model was trained on 25 manually annotated 3DMRI volumes to segment the femur, tibia, and patella from 47 078 3D MRI volumes. Each bone segmentation was converted to a 3D point cloud and transformed into spherical coordinates. Different fusion strategies were performed to merge spherical maps obtained by each bone. A total of 41 822

merged spherical maps with corresponding Kellgren-Lawrence grades for radiographic OA were used to train a CNN [10] classifier model to diagnose OA using bone shape learned features. Several OA Diagnosis models were tested and the weights for each trained model were transferred to the OA Incidence models. The OA incidence task consisted of predicting OA from a healthy scan within a range of eight time points, from 1 y to 8 y. The validation performance was compared and the test set performance was reported.

**Title:** Discriminative Regularized Auto-Encoder for early detection of knee osteoarthritis: data from the osteoarthritis initiative

**Description:** osteoarthritis (OA) is the most common disorder of the musculoskeletal system and the major cause of reduced mobility among seniors. The visual evaluation of OA still suffers from subjectivity. Recently, Computer Aided Diagnosis (CAD) systems based on learning methods showed potential for improving knee OA diagnostic accuracy. However, learning discriminative properties can be a challenging task, particularly when dealing with complex data such as X-ray images [9], typically used for knee OA diagnosis. In this paper, we introduce a Discriminative Regularized Auto Encoder (DRAE) that allows us to learn both relevant and discriminative properties that improve the classification performance. More specifically, a penalty term, called discriminative loss, is combined with the standard Auto-Encoder training criterion. This additional term aims to force the learned representation to contain discriminative information. Our experimental results on data from the public multi centre osteoarthritis Initiative (OAI) show that the developed method presents potential results for early knee OA detection.

**Title:** Exploring deep learning capabilities in knee osteoarthritis case study for classification

**Description:** This research study is devoted to the investigation of deep neural networks (DNN) for classification of the complex problem of knee osteoarthritis diagnosis. Osteoarthritis (OA) is the most common chronic condition of the joints revealing a variation in symptoms' intensity, frequency and pattern. The convolutional neural network (CNN) is a feedforward neural network that extracts deep features of images and is widely used in classification [4], object detection [5], and semantic segmentation [6].

The main goal of this research study is to implement deep neural networks as a new efficient machine learning approach for this classification task taking into account the large number of medical factors affecting OA. [17] The potential of the proposed methodology was demonstrated by classifying different subgroups of control participants from self-reported clinical data and providing a category of knee OA diagnosis. The investigated subgroups were defined by gender, age and obesity. Furthermore, to validate the proposed deep learning methodology, a comparison analysis between the proposed DNN and some benchmark machine learning techniques recommended for classification was conducted and the results showed the effectiveness of deep learning in the diagnosis of knee OA.

**Title:** Applying densely connected convolutional neural networks for staging osteoarthritis severity from plain radiographs

**Description:** Osteoarthritis (OA) classification in the knee is most commonly done with radiographs using the 0-4 Kellgren Lawrence (KL) grading system where 0 is normal, 1 shows doubtful signs of OA, 2 is mild OA, 3 is moderate OA, and 4 is severe OA. KL grading is widely used for clinical assessment and diagnosis of OA, usually on a high volume of radiographs, making its automation highly relevant [7]. We propose a fully automated algorithm for the detection of OA using KL gradings with a state-of-the-art neural network. Four thousand four hundred ninety bilateral PA fixed-flexion knee radiographs were collected from

the Osteoarthritis Initiative dataset (age = 61.2 ± 9.2 years, BMI = 32.8 ± 15.9 kg/m2, 42/58 male/female split) for six different time points. The left and right knee joints were localized using a U-net model.

These localized images were used to train an ensemble of DenseNet neural network architectures for the prediction of OA severity. This ensemble of DenseNets' testing sensitivity rates of no OA, mild, moderate, and severe OA were 83.7, 70.2, 68.9, and 86.0% respectively. The corresponding specificity

rates were 86.1, 83.8, 97.1, and 99.1%. In their study [11], they investigated numerous methods of identifying knee OA. There are three types of methods: vision-based, sensor-based, and hybrid methods. [12] used machine learning supervised classifiers such as KNN, SVM, and random forests to detect knee OA on the KL grading scale, with KNN providing high accuracy. Using saliency maps, we confirmed that the neural networks producing these results were in fact selecting the correct osteoarthritic features used in detection. These results suggest the use of our automatic classifier to assist radiologists in making more accurate and precise diagnosis with the increasing volume of radiographic images being taken in the clinic. The networks are trained using Focal Loss [18] to handle the imbalanced class.

**CHAPTER 3** **SYSTEM ANALYSIS**

**3.SYSTEM ANALYSIS**

* 1. **EXISTING SYSTEM**

The current situation has led to patients waiting for their X-rays to be manually analyzed by a radiologist. Having to analyze the X-rays of multiple patients and prepare a report for each one of them manually consumes a huge amount of time. As the workload increases, it increases the amount of time required for processing reports, which can even lead to a greater margin of error.

**Limitations of Existing System**

* + 1. In some cases, the results may be inaccurate as a result of the method.
    2. It requires more time for the task to be processed as a result.
  1. **PROPOSED SYSTEM**

Based on the anomalies in the existing system, a computerization of the entire activity is being suggested as a result of preliminary analysis. The proposed system is built using the Django Framework with Python as the programming language. It is intended that the proposed system is accessible by two entities namely the administrator and the user. In order to use the android application, the administrator needs to login with his or her valid login credentials. Once an admin is successful in logging into the system, he can access all the modules and carry out/manage all the tasks in an accurate manner.

* 1. **FEASIBILITY STUDY**

A feasibility study is a highlevel summary version of the entire process intended to answer a variety of questions, such as: What is the problem? Is there any feasible solution to the given problem? Is the problem even worth solving? It is only after a clear understanding of the problem that a feasibility study can be conducted. In order to determine whether a proposed system is

feasible, technical, operational, and economic factors must be considered. The management will have a clearer picture of the proposed system by conducting a detailed feasibility study.

To ensure that the project is variable and it does not have any major obstacles, the following options are being considered in order to make certain that it is not dependent upon any major factors.

The feasibility study consists of the following components:

* + 1. Technical Feasibility
    2. Economic Feasibility
    3. Operational Feasibility

During this phase of the process, we are going to analyze all of the proposed systems and pick the most feasible solution to the problem. We study the feasibility of a project based on three main factors, which are listed below.

* + 1. **TECHNICAL FEASIBILITY**

In order to determine whether the proposed systems are technically feasible or not, we perform this step. It refers to whether or not all the technologies required to develop the system are readily available or not. Technical Feasibility refers to whether or not the organization is equipped with the computing power and skills necessary to carry out the project. As a result of the following reasons, the system can be considered feasible for the following reasons:

* + - 1. In order for the system to be developed, all the necessary technology is available.
      2. There is too much flexibility in this system, and it can be further expanded if need be.
      3. Using this system, you will be able to guarantee the accuracy, ease of use, reliability and the security of your data.
      4. There is an instant response to inquires that can be provided by this system.

Due to the availability of all the technology needed for the project is available so project is technically feasible.

**Operating System :** Windows 7 or higher

**Languages :** Python **Database System :** My SQL **Documentation Tool :** MS – Word

* + 1. **ECONOMIC FEASIBILITY**

The project is fully feasible economically because it does not require any additional financial investment on the part of the client, and in terms of its time frame, the project could be completed within 6 months provided the correct funding is in place. The purpose of this step is to determine which of the two proposals is more economically valuable on an individual level. A comparison is made between the financial benefits of the new system and the investment that will be required to implement it. A new system can only be considered economically feasible if the financial benefits are greater than the investment and expenditures required to implement it.

Depending on the level of Economic Feasibility, we can determine whether or not the project goal is capable of being achieved within the resources allocated to it. There is a need to determine whether or not it is worthwhile to process the whole project or if the benefits gained from the new system are not worth the costs involved in implementing it. A financial benefit must be equal to or greater than the cost of a project. In this issue, we should consider:

* + - 1. An investigation of the entire system will cost a considerable amount of money.
      2. It is important to determine the cost of hard disks and software for the class of application you are considering.
      3. It is a tool that is used for the development process.
      4. It is important to consider the cost of maintenance, etc.

According to our analysis, our project is economically feasible due to the fact that the development costs are very minimal in comparison to the benefits the application will bring to the user.

* + 1. **OPERATIONAL FEASIBILITY**

During this step, we verify different operational factors of the proposed systems, such as manpower, time, etc., whichever solution uses less operational resources is the most operationally feasible. Furthermore, the solution should also be able to be implemented in an operationally feasible manner. The Operational Feasibility is a measurement of whether a proposed system would be feasible to operate within the current operating system if it satisfied user objectives:

* + - 1. Clients accept the methods for processing and presenting data because they are able to meet all their needs.
      2. A client-involved planning and development process has been followed.
      3. There will be no problem with the proposed system under any circumstances.

We consider our project to be operationally feasible because we are able to meet the time requirements as well as the personnel requirements. There are three members of our team and we worked on this project for three months over the course of the project.

* 1. **HARDWARE ENVIRONMENT**
* I5 Processor Based Computer or higher
* Memory:8 GB
* Hard Drive: 25 GB
* Monitor
* Internet Connection
  1. **SOFTWARE ENVIRONMENT**
* Windows 10 or higher
* XAMPP Server
* Sublime Text
* Python
* Google Chrome Browser or Edge Browser
  1. **ADVANTAGES AND DISADVANTAGES ADVANTAGES**

1. It helps to detect Osteoarthritis in knees instantly by detecting small changes their movement.
2. In this system, you can even find out how severe your arthritis is.
3. It is very easy to use.

**DISADVANTAGES**

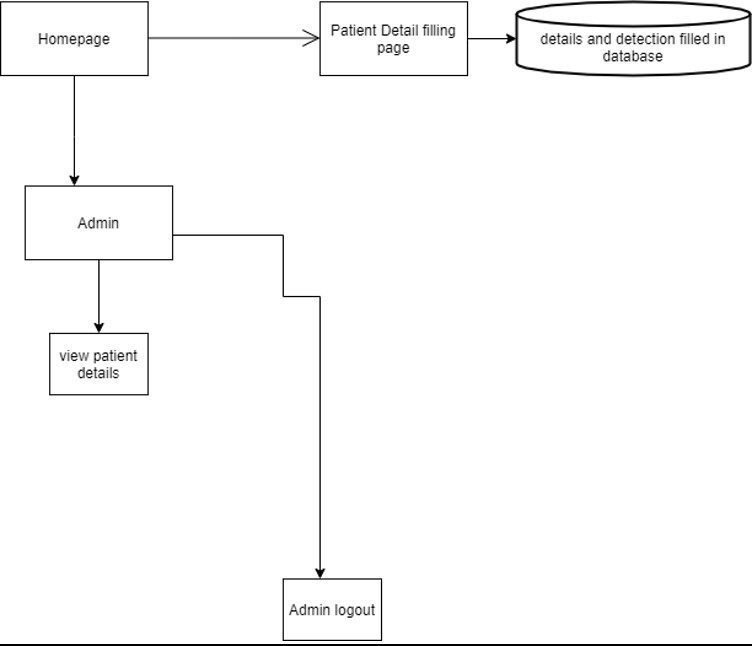
1. The sample size is small or the data is not reliable due to limited sample size.
2. Depending on the source of the data, the quality and format of the data can also vary.
3. There is a possibility of data manipulation, which could result in an incorrect prediction being made.

**CHAPTER 4** **SYSTEM DESIGN**

**4.SYSTEM DESIGN**

* 1. **SYSTEM ARCHITECTURE DIAGRAM**

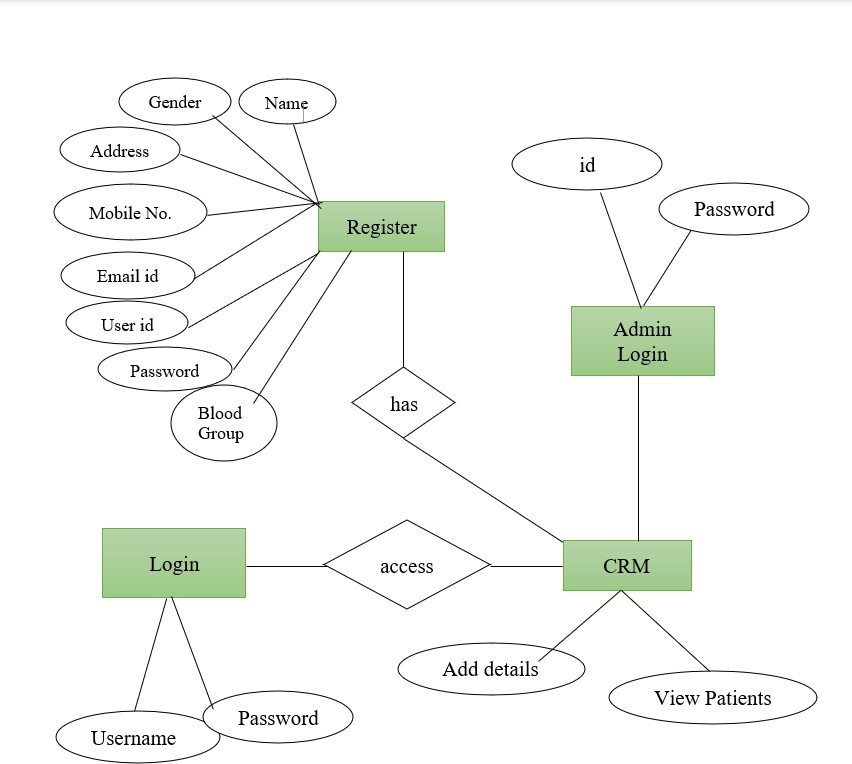
An architecture diagram for a software system is a diagram that abstracts the relationships, restrictions, and boundaries between each component within the system. A software system deployment and development roadmap is one of the most important elements of the development plan as it provides an overview of the software system's physical deployment.



**Fig 4.1 System Architecture For Admin and Patient**

* 1. **E-R DIAGRAM**

Entity relationship diagrams (ERD) are graphical representations that show relationships between people, objects, places, concepts or events within an information technology (IT) system, also known as entity relationship models or entity relationships. Using rectangles, diamonds, ovals, as well as connecting lines and rectangles, they demonstrate how entities, relationships, and attributes are interconnected through the use of shapes and shapes.



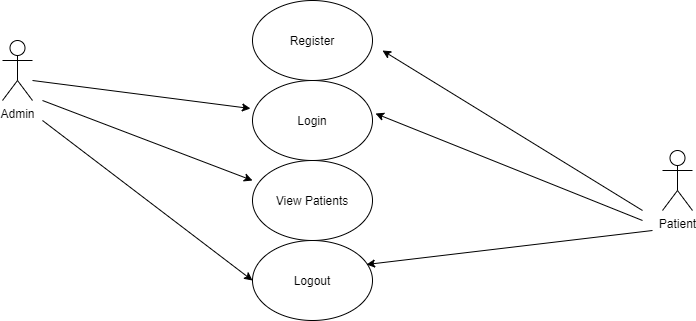
**Fig 4.2 E-R Diagram for Whole System**

* 1. **USE CASE DIAGRAM**

The Unified Modeling Language (UML) is a general-purpose modeling language used in software engineering. Object Management Group is responsible for managing and creating the standard. A set of graphic notation techniques is included in UML to create visual models of software-intensive systems. Object- oriented software intensive systems are specified, visualized, modified, constructed, and documented using this language. A use case diagram is a visual representation of the functionality provided by a system as a set of actors, their goals, and the dependencies between those actors and use cases that relate to the system.

The use case diagram is divided into two parts:

* + 1. **Use case:** Use cases describe sequences of actions that provide something of measurable value to one or more actors, and they are characterized by the presence of horizontal ellipses that represent them.
    2. **Actor:** An actor is an individual, organization, or system that holds a particular position, a function, or a role in interaction with a system.



**Fig 4.3 Use Case Diagram for Admin and Patient**

* 1. **CLASS DIAGRAM**

A Class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

**Class Notations**

* + 1. Class Name

The first partition contains the name of the class.

* + 1. Class Attributes

The second partition shows attributes. After the colon, the attribute type is shown. In code, attributes correspond to member variables.

* + 1. Class Operations (Methods)

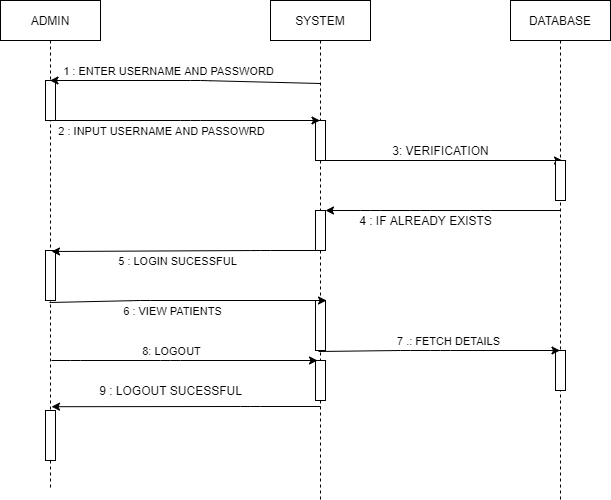
The third partition shows operations. These are the services provided by the class. When the method signature ends with a colon, the return type appears. Parameter names are followed by a colon indicating the return type. Class methods correspond to operations in code.



**Fig 4.4 Class Diagram for User and New Registration**

* 1. **SEQUENCE DIAGRAM**
     1. **ADMIN SEQUENCE DIAGRAM**

A sequence diagram is a type of interaction diagram that shows how processes work together as well as in what sequence they occur. The Message Sequence Diagram is a construct found within Message Sequence Diagrams, which are also sometimes referred to as event diagrams, event sceneries, and timing diagrams.

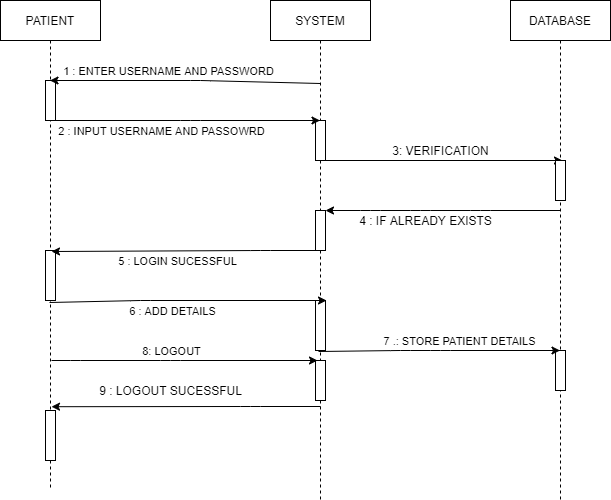


**Fig 4.5.1 Admin Sequence Diagram**

* + 1. **PATIENT SEQUENCE DIAGRAM**

The purpose of a sequence diagram is to show the flow of events

* + - 1. The purpose of this model is to show how high-level interactions occur between active objects within a system.
      2. It is the goal of the project to model interaction between objects inside a collaboration that is based on a use case.
      3. Depending on what kind of interaction it is modelling, it could either be a generic interaction or a certain kind of interaction.

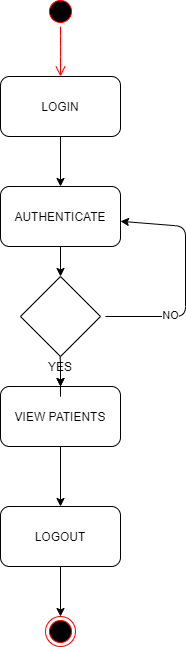


**Fig 4.5.2 Patient Sequence Diagram**

* 1. **ACTIVITY DIAGRAM**

The activity diagram represents the workflow of a series of stepwise activities and actions in a graphical manner. It supports choice, iteration, and concurrency within the workflow. An activity diagram illustrates the flow of control.

* + 1. **ADMIN ACTIVITY DIAGRAM**

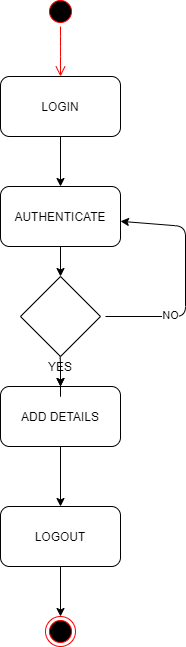


**Fig 4.6.1 Admin Activity Diagram**

* + 1. **PATIENT ACTIVITY DIAGRAM**

Shapes can be categorized into the following types:

* + - 1. Activities are represented by rounded rectangles.
      2. Decisions are represented by diamonds.
      3. The bars indicate the start and end of concurrent activities.
      4. A black circle starts the workflow. An encircled circle ends the workflow



**Fig 4.6.2 Patient Activity Diagram**

* 1. **DATA FLOW DIAGRAM**

Data flow diagrams are diagrams that display the movement and analysis of business data through a system in a graphical format. This is the central tool and the foundation from which all the other components are developed, as they are designed around this tool. There are multiple ways in which the transformation of data from input to output, through processing, can be described logically and independently of the physical components associated with the system. A logical data flow diagram is a diagram that represents the flow of data in a logical manner. The physical data flow diagrams illustrate how data is moved between people, departments, and workstations in the context of the actual implements and movements of the data.

* + 1. **DATABASE DETAIL**

Query

0.0

Knee

Osteoarthritis Detection DB

Database

User

**Fig 4.7.1 Database Details of User**

* + 1. **LEVEL 1 DFD**

Query

User

1.0

Process

Check for user Requirement

User need

Feedback For User

1.1

Relevant Data

Database

* + 1. **LEVEL 2 DFD**

**Fig 4.7.2 Level 1 DFD**

Query

2.0

User

Accept

Give request to user

Query

2.1

Check Availability of or for query processing

Give info about DB

2.2

Process Query

Via Knee Osteoarthritis Detection DB

**Fig 4.7.3 Level 2 DFD**

**CHAPTER 5**

**SYSTEM IMPLEMENTATION**

**5.SYSTEM IMPLEMENTATION**

* 1. **MODULES AND ITS DESCRIPTION**
     1. **ADMIN MODULE**

An administrator can use the admin module to set up the back-end of the system and to perform basic configuration of the system. In order to check the details of the entered data for a patient, the admin will connect with the database and check the details of that data and provide severity details.

The admin module contains mainly 3 modules:

* + - 1. **Login**- The administrator can log into the system by entering their username and password in the appropriate fields.
      2. **View Patient** - The admin has the ability to view the details of the patient.
      3. **Logout** – Administrators can log out of the system.
      4. **LOGIN**

A username and password need to be created in the SQL database by the admin. The admin will need to start the XAMPP control panel, start Apache, and start the SQL database. A user will need to click on the admin button on the SQL server XAMPP server. This will open the localhost site, and the knee database can be found on the left side of the screen. Once you have clicked on the admin details, the username and password need to be inserted.

* + - 1. **VIEW PATIENT**

Once you log into your admin account, you will find a text link that says "See Patient " at the top of the page on the left side. On clicking it, you will be able to view the patient details such as Name, Age, Symptoms, Detection, Username, solution and Images.

* + - 1. **LOGOUT**

An admin can log out at any time by clicking the "Logout" button that will be available at the top of the page on the right.

* + 1. **PATIENT MODULE**

The term patient refers to anyone who receives health care services that are performed by health care professionals. It is most often the case that the patient has been injured or is ill and needs medical attention from a physician, nurse, optometrist, dentist, veterinarian, or another health care professional.

The patient module contains mainly 2 modules:

* + - 1. **Login** – The sign-in button is available for patients to use to log into the system.
      2. **Register**- It is possible for patients to register themselves in the system in order to predict their disease in advance.
      3. **LOGIN**

The patient can login into his account by just clicking on the sign-in button on the top right of the page. It will then display the patient and then they will need to click on it in order to do so.

* + - 1. **REGISTER**

After the patient login, the patient needs to enter the details that contain the Name, Age, Previous History, Username, Password, DATE of birth, Blood group, X ray photo, your photo, your photo, Guardian photo, ID photo, License photo, and click the go button. Once the patient has done this, a pop-up message will appear.

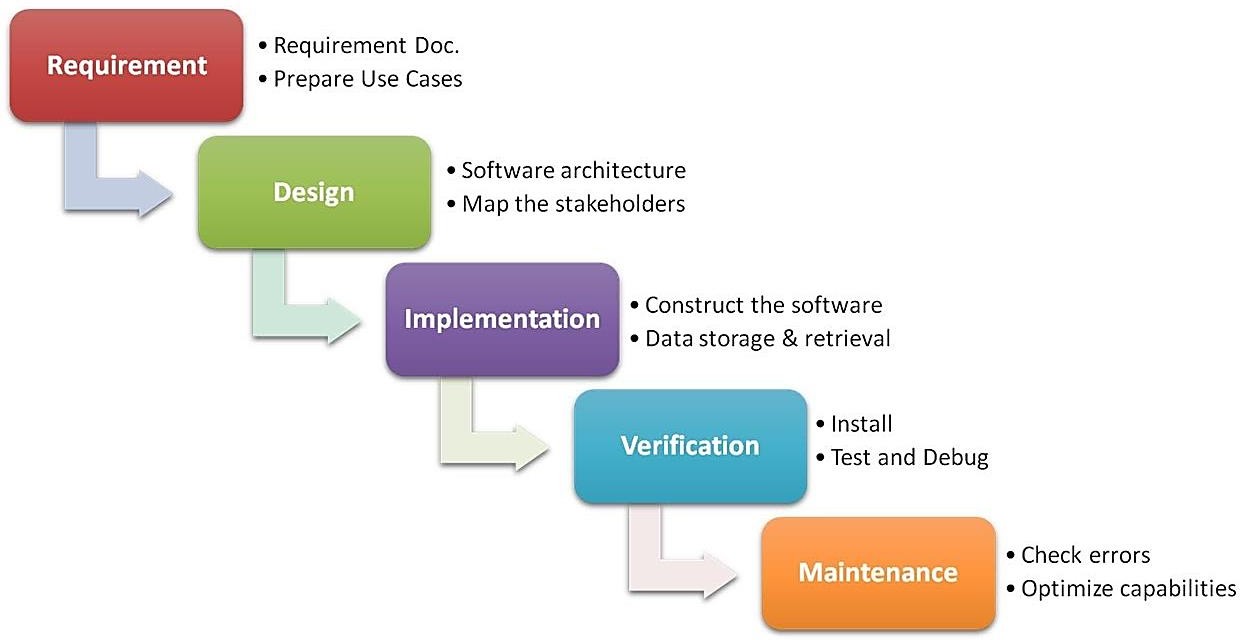
* 1. **PROJECT IMPLEMENTATION TECHNOLOGY**

This project has been loaded into Visual XAMPP Server & Sublime Text. It was used to design and code the project using that software. We have created and maintained all databases in MYSQL, in which we created tables and wrote queries to store data or keep records about the project.

* + 1. **PROJECT LIFECYCLE DETAILS**

A Project Lifecycle is a series of phases through which a project advances during its development. According to the type of project and the type of company involved, there may be a difference in the number of phases or the sequence of phases within the cycle. However, in order for them to be considered part of a project, they should be able to have a definite beginning and an end, and they need to be time-bound. It is essential to recognize that the life cycle provides the foundation for the actions that have to be completed in a project, regardless of the specifics that are involved in the project.

* + - 1. **WATERFALL MODEL**



**Fig 5.2.1.1 Waterfall Model**

The waterfall model describes the flow of information in a linear sequential manner. Through software implementation phases, progress is seen as flowing steadily downwards (like a waterfall) through the phases of the development process. It means that any phase in the development process can only be initiated if the previous phase has been completed before the next phase can begin. In the waterfall approach, there is no defined process to go back to the previous phase

in the event that requirements change during the process. For the development of software, the waterfall approach was the earliest method that was used for the development of software.

* + 1. **TECHNOLOGY USED**
       1. **PYTHON**

Guido van Rossum has created a powerful programming language called Python, which is a versatile multipurpose language. I would describe Python as a general- purpose programming language. A wide range of applications can be developed with Pygame, from Web development (like: Django and Bottle) to scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user interfaces (Pygame, Panda3D). It is a clean-looking language with a relatively short amount of code, and the syntax of the language is also very clean. Python is a great language to work in since it allows you to focus on the problem you are trying to solve rather than the syntax you are trying to follow.

Python programming has the following features:

* + - * + It is a simple language that is easier for beginners to learn

The syntax of Python is very simple and elegant, so it is very easy to learn. Compared to other languages like C++, Java, C#, Python is much easier to read and write than other programming languages such as: C++, Java, and C#. In addition to making programming fun, Python allows you to focus on the solution rather than the syntax when writing code.

* + - * + The software is free and open-source

The Python programming language can be freely used and distributed, even for commercial purposes. Aside from being able to use and distribute software written in Python, you can also make changes to Python's source

code, which is constantly being improved by the Python community with every new release.

* + - * + It is portable

Python is a lightweight programming language that can run on almost all of the major platforms, including Windows, MacOS X and Linux, and you can easily move Python programs between them.

* + - * + An object-oriented programming language

Python is an object-oriented language. Object-oriented programming (OOP) helps you to solve a complex problem in an intuitive manner. By creating objects, you are able to divide this complex problem into smaller sets, which in turn makes it easier for you to solve it.

* + - * + It is a high-level, interpretive language that can be used by anybody

The biggest advantage of Python over C/C++ is that you do not have to worry about daunting tasks like memory management, garbage collection, etc. Also, when you run Python code, it automatically converts your code into a language your computer can understand. It does not require you to worry about any operations at the lower levels of the system.

* + - 1. **DJANGO**

Django is a high-level Python Web framework that promotes rapid development and clean, pragmatic designs. This software, which is built by experienced developers, takes care of many of the hassles associated with Web development, so you can focus on writing your app instead of reinventing the wheel every time. The software is open source and free to use.

**Features of Django**

* + - * + Rapid Development
        + Secure
        + Scalable
        + Fully loaded
        + Versatile
        + Open Source
        + Vast and Supported Community

**Rapid Development**

It was specifically designed to make a framework that would make it faster to build web applications than others, which is why Django was designed. It is a very time-consuming process to implement a project, but Django makes this process as quick as possible.

**Secure**

The Django framework takes security very seriously and enables developers to avoid many common security mistakes, such as SQL injections, cross-site scripting attacks, cross-site request forgeries, and the like. The user authentication system it uses makes it possible for users to manage their passwords and accounts in a secure way.

**Scalable**

There are a variety of features that make Django scalable, and you are able to switch from small-scale application projects to large-scale projects quickly and easily.

**Fully loaded**

In Django, there are a variety of helping task modules as well as libraries which can be used to perform common Web development tasks. Authentication of users, content administration, site maps, RSS feeds, and so on are all handled by Django as part of its functionality.

**Versatile**

It is a flexible language which allows it to build applications for a range of different domains owing to its versatility. There are several types of applications that are being built using Django nowadays, such as content management systems, social network platforms and scientific computing platforms.

**Open Source**

Django is an opensource framework for building web applications. In addition to being publicly available, it is also free of charge. The application is open source, which means it can be downloaded with the source code from the public repository. Open source reduces the total development cost of the application.

**Vast and Supported Community**

There are many web frameworks out there, but Django is one of the most popular. There is a large community of users who support the platform, as well as channels for sharing and connecting.

* + - 1. **XAMPP SERVER**

There is a software program called XAMPP that is designed to install and configure Apache web server, MySQL database server, PHP scripting language, phpMyAdmin (to manage MySQL database), and SQ Lite Manager (to manage SQLite database) under Windows OS. XAMPP is designed to provide an easy way to install Apache, PHP, and MySQL packages with an easy-to-use

installation program, instead of having to configure the packages yourself after you have installed and configured them.

It's so easy to get started with XAMPP because once you've installed it, you're ready to go. If you want to get it running, you don't have to do any additional configuration or tweaking of any configuration files to achieve that. There are usually two reasons why someone chooses to install XAMPP on their computer. For development purposes or for running their own server, they are interested in installing XAMPP for development purposes.

**XAMPP Server Contains PHP Admin**

The phpMyAdmin is a free PHP based software tool that allows you to edit or create new users, and to create new databases, and is intended to be used over the internet in order to manage MySQL databases. There are a wide variety of MySQL operations that can be performed using phpMyAdmin. You will have the ability to manage databases, tables, fields, relations, indexes, users, permissions, etc., via the user interface, while still having the ability to directly execute any SQL statement that you may need (such as managing databases, tables, fields, relations, indexes, users, permissions, etc.).

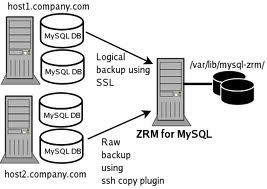
**Features**

* Web interface that is intuitive and easy to use
* MySQL features supported:

1. Databases, tables, views, fields, and indexes can be browsed and dropped.
2. You can create, copy, drop, rename, and alter databases, tables, fields, and indexes.
3. Providing proposals for the configuration of the server, databases, and tables.
4. You can execute, edit, and bookmark any SQL statement, even batch queries.
5. The management of MySQL users and privileges is important
6. Manage stored procedures and triggers in the database.

* Data can be imported from CSV and SQL files
* It is possible to export data in a wide variety of formats such as CSV, SQL, XML, PDF, ISO/IEC 26300 - OpenDocument Text and Spreadsheet, Word, LaTeX, and others.
* It is possible to administer multiple servers at the same time
* You can create PDF graphics of your database layout by creating a PDF document
* Query-by-Example (QBE) is a powerful tool for creating complex queries
* It is possible to search globally in a database or a subset of a database
* Transforming stored data into any format using a set of predefined functions, like displaying BLOB-data as an image or a link to download the data
  + - 1. **SQL SERVER**

SQL Server is a relational database management system from Microsoft that has been developed to be used in enterprise environments. This version of SQL Server is built upon the T-SQL programming language from Sybase and Microsoft, a set of extensions that extends the standard SQL language with several new features including transaction control, exception handling, row processing, and variables that are declared in the code.



**Fig5.2.2.4 SQL Server**

In general, any database management system (DBMS) which is capable of responding to queries from client machines that are formatted in the SQL language is classified in this category. The term "database management system", when capitalized, refers to either of two database management products offered by Sybase and Microsoft. DBMS products called SQL Server are offered by both companies as client-server DBMS products.

**Using XAMPP as a Development Server**

In order to develop and test websites locally on their own computer, you can simply use XAMPP instead of installing a web hosting account on your computer for the purpose of developing websites. XAMPP will be used by the majority of

people for development purposes, such as learning how to create websites with HTML, PHP, and MySQL through XAMPP.

**Using XAMPP as a Production Server**

The XAMPP server was designed to be a development and testing server, and not a live production server. Due to the fact that XAMPP does not come with any security in place, it does not provide any kind of protection against any kind of attack. A 10-year-old child with access to the internet can easily hack into your XAMPP server if he has access to the internet.

It is very important that you keep in mind when you are setting up your website(s) that they may contain highly sensitive data (such as credit card numbers, social security numbers, user ids, passwords, etc.). Before you upload this information to your website(s), you need to take this into consideration. XAMPP should never be used for a production server unless you are an experienced system administrator who is capable of configuring it to be more secure.

**MySQL Configuration**

To begin the MySQL installation process, download the latest version of Essentials as an MSI package first. In the installation process of MySQL, make sure you select the Typical installation option and use the default configuration values, with the exception of Sign-Up, where you may want to skip it. Upon the completion of the Setup Wizard, make sure that the option Configure the MySQL Server now is selected in the options list. When configuring an instance of MySQL Server, select Standard Configuration from the drop-down menu. The next thing that you need to do is to set the option Include Bin Directory in Windows PATH to Yes. The setting of this parameter is crucial, otherwise a

required library, libMySQL.dll, will not be found during the startup of Apache if this parameter is not set.

It is important to enter the correct root password at the end of the process. It is not necessary to enable remote root access nor to create an Anonymous Account if you do not wish to do so. It is recommended that you inspect messages displayed during MySQL startup and ensure that MySQL has been started successfully. Afterwards, you will need to reboot the computer in order to complete the process. The worst case scenario is for the required librarylibMySQL.dll not to be found during Apache startup when Apache is trying to load Apache's PHP module and Apache will, perhaps a bit confusingly, complain that it is unable to load the PHP's MySQL library, php\_mysql.dll, during Apache startup. This means that the system must be rebooted at this stage and then the PHP configuration can be completed.

**CHAPTER 6** **TESTING**

**6. TESTING**

* 1. **LEVELS OF TESTING**

It is through the concept of levels of testing that we are able to uncover the errors present in different phases of the development process. Testing can be divided into the following levels:

Client Needs Acceptance Testing

Requirements System Testing

Design Integration Testing

Code Unit Testing

The proposal for a new system must undergo a number of tests before the system can be ready for user acceptance testing. In the process of testing, the following steps are involved:

* + 1. **UNIT TESTING**

The goal of unit testing is to focus the verification efforts on the smallest unit of the software design, which is the module. There is also a term called "Module Testing" that applies to this process. Modules are tested separately from each other. In order to carry out this testing, it was done during the programming stage

itself. According to this testing, each module has been found to function satisfactorily as far as the expected output from the module is concerned.

* + 1. **INTEGRATION TESTING**

It is possible to have data that has been aggregated across an interface; one module can have adverse effects on another module. Testing the integration of a program is a systematic process of testing the structure of the program while simultaneously conducting tests to identify errors associated with the interface that are causing the program to behave in a particular manner. We are going to build a program structure by taking unit tested modules and putting them together in a cohesive manner.

It is important to note that all the modules have been combined and tested together as a whole. There is a difficulty in correcting the problem here, since the isolation of the cause has been complicated by the vast expenditures associated with the entire program. Therefore, in the integration testing stop, all of the errors uncovered during the text testing steps will be corrected in the integration testing step.

* + 1. **SYSTEM TESTING**

A system test is a stage of implementation that is aimed at ensuring that the system works accurately and efficiently before going live with its operation in order to ensure its long-term success. The success of the system depends on the success of the testing. When it comes to testing a system, the logical assumption is that if all the parts of the system are in order, then the goal of the system will be achieved successfully.

* + 1. **VALIDATION TESTING**

The software is now completely assembled as a package as the result of integration testing. Interfacing errors have been discovered and corrected, and a final series of software tests is about to be conducted, and then the validation test begins. There are many ways in which validation tests can be defined. In reality, the simple definition of validation is that it is accomplished when the software functions in a way that the customer can reasonably expect, which is when the software functions as expected. One of two possible outcomes will occur after a validation test has been conducted.

One of these outcomes is a function or performance characteristic that conforms to specifications and is accepted, while the other outcome is a deviation from specification that is revealed and a deficiency list is generated. A validation test has been conducted on the proposed system under consideration, and it has been found to be working satisfactorily.

* + 1. **OUTPUT TESTING**

The next step after completing the validation testing is to go through the output testing of the proposed system. In order for a system to be useful, it must produce the required output in the specified format. The outputs generated by the system for which the evaluation is being conducted are tested by asking the users about the format that they require. The output format of this document is considered from two points of view, namely on the screen as well as on paper. We have found that the format of the output on the screen is correct since the format was designed in the system design phase in accordance with the requirements of the users.

For the hard copy, as well, the output comes according to the requirements specified by the users. This means that output testing does not result in any corrections in the system as a result of the output testing.

* + 1. **USER ACCEPTANCE TESTING**

A system's success is directly proportional to the level of acceptance it receives from its users. As part of the development process of the system under study, it has been ensured that the proposed system is tested for user acceptance by keeping in continuous touch with the prospective users at the time of its development and making changes where necessary as a result.

* 1. **TEST CASES**

As far as the test cases are concerned, there are only two of them.

* Registration
* Login
  + 1. **REGISTRATION**

As a first step, users are required to register by filling out some basic registration information in order to log in. In the registration page, there are several fields that must be filled in by the user, and every field must be filled in by the user. It is not possible for the user to use a character in the login id field.

* + 1. **LOGIN**

Login id and password are kept compulsory fields, and if the id or password doesn’t match then it will show an error message.

* + 1. **VALIDATION CRITERIA**
       1. In every form, there should not be any blank fields in the fields that are not null.
       2. It is recommended that all numeric fields are checked for non-numeric values. In a similar way, text fields, such as names, should not contain any numeric characters in them.
       3. There should be a way to automatically generate all primary keys in order to prevent the user from entering any existing key.
       4. The use of error handling for every operation that is important, such as saving, editing, deleting, and other important operations.
       5. If the user tabs out or enters from a text box, the data should be validated and, if the data is invalid, focus should be sent back to the text box with a proper message as soon as the user returns to the text box.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Id | Test Case Scenario | Test Steps | Expected Result | Actual Result | Pass/Fail |
| 1 | The user filling in the text fields of registration form | We will check here if the text fields do not contain numeric values | If it contains numeric value do not accept else accept | It accepts when we provide non numeric values in text fields | Pass |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2 | The user filling in the numeric fields of registration form | We will check here if the numeric fields do not contain non- numeric values | If it contains non-numeric value do not accept else accept | It accepts when we provide numeric values in numeric fields | Pass |
| 3 | When the user is trying to login | We will check if the password and username are correct | If both are correct we will allow them to login else throw an error | It allows the user to login when the credentials are correct else throws an error | Pass |

**CHAPTER 7 CONCLUSION**

**7.CONCLUSION**

**7.1 PROJECT CONCLUSION**

Developing a system that detects knee osteoarthritis and its severity in Python programming language as part of a project called System Design on Knee Osteoarthritis Detection and Severity. There is a lot of effort that goes into the development of this system on our part. As a team, we think this system has given us all a lot of satisfaction. Despite the fact that every task in this field is never said to be perfect, there may still be room to improve on this application in the future. It was a great experience for us as we learned many things about the development field and gained a lot of knowledge about it. It is our hope that this will prove fruitful for us in the future.

**APPENDICES** **CODING**

* 1. **PROJECT CODING**

**APPENDICES**

from django.shortcuts import render,redirect from django import template

from django.contrib.sessions.models import Session import string

from datetime import date import datetime

from datetime import datetime import datetime

from datetime import date

from django.http import HttpResponse from django.db.models import Q

from django.contrib.auth.forms import UserCreationForm

from django.contrib import messages

from django.db.models import Avg, Max, Min, Sum, Count from django.views.decorators.csrf import csrf\_exempt

from django.http import JsonResponse

from .forms import ImageFileUploadForm from kneearth.models import \*

from django.shortcuts import render,redirect from django.http import HttpResponse

from tkinter import \* import tkinter.messagebox import cv2

import cv2

import numpy as np import pytesseract

from django.contrib.auth.forms import UserChangeForm, UserCreationForm from django.contrib.auth.forms import AuthenticationForm

import os

from math import hypot import numpy as np

from keras.models import model\_from\_json import operator

import cv2

import sys, os

from tkinter import \* import numpy as np import pandas as pd import re

import os

import tensorflow as tf from numpy import array

from keras.datasets import imdb

from keras.preprocessing import sequence from keras.models import load\_model

def home(request):

return render(request,'home.html',{}) def about(request):

return render(request,"about.html",{}) def Admin\_login(request):

if request.method == 'POST':

Username = request.POST['Username']

password = request.POST['password']

if Admin\_Details.objects.filter(Username=Username, Password=password).exists():

user = Admin\_Details.objects.get(Username=Username, Password=password)

request.session['type\_id'] = 'Admin' request.session['username'] = Username request.session['login'] = 'Yes'

return redirect('/')

else:

messages.info(request,'Invalid Credentials') return redirect('/Admin\_login/')

else:

return render(request, 'Admin\_login.html', {}) def Stylish\_login(request):

if request.method == 'POST':

Username = request.POST['Username'] password = request.POST['password']

if Stylish\_details.objects.filter(Username=Username, Password=password,Status='Confirmed').exists():

Sty = Stylish\_details.objects.get(Username=Username, Password=password)

request.session['Stylish\_id'] = str(Sty.id) request.session['type\_id'] = 'Stylish' request.session['username'] = Username request.session['login'] = 'Yes'

return redirect('/') else:

messages.info(request,'Invalid Credentials') return redirect('/Stylish\_login/')

else:

return render(request, 'Stylish\_login.html', {}) def User\_login(request):

if request.method == 'POST':

Username = request.POST['Username'] password = request.POST['password']

if User\_details.objects.filter(Username=Username,

Password=password).exists(): ans = ""

users = User\_details.objects.all().filter(Username=Username) ExpiryDate = users[0].ExpiryDate print("ExpiryDate",ExpiryDate)

Curr\_date = datetime.datetime.now() Cyear = Curr\_date.strftime('%Y') Cmon = Curr\_date.strftime('%m') Cday = Curr\_date.strftime('%d') print("Curr\_date",Curr\_date)

Eyear = ExpiryDate.strftime('%Y') Emon = ExpiryDate.strftime('%m') Eday = ExpiryDate.strftime('%d')

ex\_date = date(int(Eyear), int(Emon), int(Eday)) curr\_date = date(int(Cyear), int(Cmon), int(Cday)) delta = ex\_date - curr\_date

if ex\_date > curr\_date: print("Greater")

else:

print("Lesser") diff = delta.days if int(diff) > 0:

ans = "Yes" else:

ans = "No"

if ans == "Yes":

user = User\_details.objects.all().filter(Username=Username, Password=password)

request.session['User\_id'] = str(user[0].id) request.session['type\_id'] = 'User' request.session['username'] = Username

request.session['login'] = 'Yes' User\_details.objects.filter(id=str(user[0].id)).update(ChatStatus='Online')

return redirect('/') else:

messages.info(request,'You Subscriptions is over') return redirect('/User\_login/')

else:

messages.info(request,'Invalid Credentials') return redirect('/User\_login/')

else:

return render(request, 'User\_login.html', {}) def Register(request):

if request.method == 'POST':

first\_name = request.POST['first\_name'] last\_name = request.POST['last\_name'] Username = request.POST['Username'] Email = request.POST['Email']

Mobile = request.POST['Mobile'] Password = request.POST['Password'] Address = request.POST['Address'] State = request.POST['State']

City = request.POST['City']

CreditCard = request.POST['CreditCard'] MM = request.POST['MM']

YYYY = request.POST['YYYY'] CCExpiry = MM+"/"+YYYY Cvv = request.POST['Cvv']

Expiry\_date = datetime.datetime.now() + datetime.timedelta(30) Expiry\_date = Expiry\_date.strftime('%Y-%m-%d') print(Expiry\_date)

if User\_details.objects.filter(Username=Username).exists(): messages.info(request,'Username taken')

return redirect('/AddOfficer/')

elif User\_details.objects.filter(Email=Email).exists(): messages.info(request,'Email Id taken')

return redirect('/AddOfficer/') else:

register1 = User\_details( FirstName = first\_name,LastName = last\_name,Username =Username,Email = Email,Password = Password,mobile = Mobile,Address = Address,City = City,State = State,ExpiryDate=Expiry\_date,CreditCard = CreditCard,Expiry = CCExpiry,Cvv = Cvv,ChatStatus='Offline')

register1.save()

messages.info(request,'Registration Done Successful')

return redirect('/Register/')

else:

return render(request, 'register.html', {}) def StylishRegister(request):

if request.method == 'POST': name = request.POST['name']

contact = request.POST['contact'] emailid = request.POST['emailid'] Username = request.POST['Username'] Password = request.POST['Password'] Address = request.POST['Address'] Image1 = request.FILES['Image1'] Image2 = request.FILES['Image2'] Image3 = request.FILES['Image3'] Image4 = request.FILES['Image4'] Image5 = request.FILES['Image5'] Speciality = request.POST['Speciality']

if Stylish\_details.objects.filter(Username=Username).exists():

messages.info(request,'Username taken') return redirect('/StylishRegister/')

elif Stylish\_details.objects.filter(Email=emailid).exists(): messages.info(request,'Email Id taken')

return redirect('/StylishRegister/') else:

register1 = Stylish\_details( Name = "Name",Contact = "Age",Address = "Symptoms",Email = "Detection",Username = "Username ",Password = "Password",Image1 = Image1,Image2= Image2,Image3 = Image3,Image4 = Image4,Image5 = Image5,Status = "Pending",Speciality="Solution")

register1.save() print("abc") print(Image1)

books\_image = str(Image1) books\_image = books\_image books\_image = books\_image print(name)

img\_name = "C:/workspace/Project/knee\_arthiritis/media/img/images/"

+ books\_image

img\_name=img\_name

img\_name=img\_name print(img\_name)

json\_file = open("C:/workspace/Project/knee\_arthiritis/kneearth/model- bw.json", "r")

model\_json = json\_file.read() json\_file.close()

loaded\_model = model\_from\_json(model\_json)

# load weights into newmodel loaded\_model.load\_weights("C:/workspace/Project/knee\_arthiritis/kneearth/mo del-bw.h5")

print("Loaded model from disk") a = cv2.imread(img\_name)

a = cv2.flip(a, 1)

roi = cv2.resize(a, (64, 64))

roi = cv2.cvtColor(roi, cv2.COLOR\_BGRA2GRAY)

\_, test\_image = cv2.threshold(roi, 100, 255, cv2.THRESH\_BINARY) result = loaded\_model.predict(test\_image.reshape(1, 64, 64, 1)) print(result)

vector = np.array([test\_image.flatten()])

print(vector)

prediction = {'INITIAL STAGE': result[0][0], 'LOW': result[0][1],

'MEDIUM': result[0][2],

'HIGH': result[0][3],

'CRITICAL': result[0][4]

}

# Sorting based on top prediction

prediction = sorted(prediction.items(), key=operator.itemgetter(1), reverse=True)

print(prediction) a = prediction

a = (a[0])

a = a[-2]

print(a) a = a

a = a

if a== "INITIAL STAGE":

Address = "The joint may become stiff and swollen, making it difficult to bend and straighten the knee.Pain and swelling may be worse in the morning, or after sitting or resting."

Speciality = "Weight loss and Exercise" print(Address)

elif a== "LOW":

Address = "Patients develop very minor wear & tear and bone spur growths at the end of the knee joints."

Speciality = "Accessories,Braces and wraps can help stabilize your

knee"

print(Address)

elif a== "MEDIUM":

Address = "knee joints will show more bone spur growth, and though the space between the bones appear normal, people will begin experiencing symptoms of joint pain"

Speciality = "knee wraps. acetaminophen. nonsteroidal anti- inflammatory drugs (NSAIDs) nonsteroidal gels"

print(Address) elif a== "HIGH":

Address = "With the progression of osteoarthritis of the knee, there is obvious joint inflammation which causes frequent pain when walking, running, squatting, extending or kneeling"

Speciality = "you should continue with nonpharmacological therapies such as exercise and weight loss. People with stage 3 OA will also continue to receive NSAIDs or acetaminophen."

print(Address)

elif a== "CRITICAL":

Address = "the joint space between the bones are considerably reduced, causing the cartilage to wear off, leaving the joint stiff. The breakdown of cartilage leads to a chronic inflammatory response, with decreased synovial fluid that causes friction, greater pain and discomfort when walking or moving the joint"

Speciality = "Soft tissue destruction may be noted around the knees in this stage.Treatment options include osteotomy or bone realignment surgery"

print(Address)

books\_image = "C:/workspace/Project/knee\_arthiritis/media/img/images/"

+ books\_image

print("abc") abc = "abc" print(abc) print(Address)

Address = Address

register1 = Stylish\_details( Name = name,Contact = contact,Address = Address,Email = Username + " has been detected with " + a + " condition

",Username = Username,Password = Password,Image1 = Image1,Image2= Image2,Image3 = Image3,Image4 = Image4,Image5 = Image5,Status = "Pending",Speciality=Speciality)

register1.save()

messages.info(request,'Details filled! See your results in the admin page!') return redirect('/StylishRegister/')

else:

return render(request, 'StylishRegister.html', {}) def logout(request):

if request.session['type\_id'] == 'User': uid = request.session['User\_id']

User\_details.objects.filter(id=uid).update(ChatStatus='Offline') elif request.session['type\_id'] == 'Stylish':

uid = request.session['Stylish\_id'] User\_details.objects.filter(id=uid).update(ChatStatus='Offline')

Session.objects.all().delete() return redirect('/')

def ViewUsers(request):

if request.method == 'POST':

return redirect('/ViewUsers/') else:

users = User\_details.objects.all() print(users)

return render(request, 'ViewUsers.html', {'users':users}) def Stylish\_Chat(request):

if request.method == 'POST': return redirect('/Stylish\_Chat/')

else:

Sid = request.session['Stylish\_id']

ids = Chat\_details.objects.filter(Sid=Sid).values\_list('Uid', flat=True) print('ids',ids)

users =

User\_details.objects.all().filter(ChatStatus='Online').filter(id in=ids) print(users)

return render(request, 'Stylish\_Chat.html', {'users':users}) def User\_Chat(request):

if request.method == 'POST':

return redirect('/User\_Chat/')

else:

sty = Stylish\_details.objects.all()

return render(request, 'User\_Chat.html', {'sty':sty}) def MyDetails(request):

if request.method == 'POST': return redirect('/MyDetails/')

else:

uid = request.session['User\_id']

users = User\_details.objects.all().filter(id=uid) ExpiryDate = users[0].ExpiryDate print("ExpiryDate",ExpiryDate)

Curr\_date = datetime.datetime.now() Cyear = Curr\_date.strftime('%Y') Cmon = Curr\_date.strftime('%m') Cday = Curr\_date.strftime('%d') print("Curr\_date",Curr\_date)

Eyear = ExpiryDate.strftime('%Y') Emon = ExpiryDate.strftime('%m')

Eday = ExpiryDate.strftime('%d')

ex\_date = date(int(Eyear), int(Emon), int(Eday)) curr\_date = date(int(Cyear), int(Cmon), int(Cday)) delta = ex\_date - curr\_date

if ex\_date > curr\_date: print("Greater")

else:

print("Lesser") diff = delta.days ans = "Yes"

if int(diff) <= 5: ans = "Yes"

else:

ans = "No"

return render(request, 'MyDetails.html', {"users":users,"ans":ans}) def UpdateSubscription(request):

if request.method == 'POST': UserId = request.POST['UserId']

if User\_details.objects.filter(id = UserId).exists(): Package = request.POST['Package'] CCNumber = request.POST['CCNumber'] Month = request.POST['Month']

Year = request.POST['Year'] Cvv = request.POST['Cvv'] CCExpiry = Month+"/"+Year count = 0

if Package == "Half Yearly": count= 182

elif Package == "Yearly": count= 365

users = User\_details.objects.all().filter(id=UserId) ExpiryDate = users[0].ExpiryDate

Expiry\_date = ExpiryDate + datetime.timedelta(int(count)) Expiry\_date = Expiry\_date.strftime('%Y-%m-%d') print(Expiry\_date)

OriginalCC = users[0].CreditCard

OriginalExp = users[0].Expiry OriginalCvv = users[0].Cvv

if OriginalCC == CCNumber and OriginalCvv == Cvv and OriginalExp

== CCExpiry:

User\_details.objects.filter(id=UserId).update(ExpiryDate=Expiry\_date) messages.info(request,'Plan updated Successfully')

else:

messages.info(request,'Credentials doesnt match')

else:

messages.info(request,'Id doesnt match') return render(request, 'MyDetails.html', {})

else:

return redirect('/') def ViewStylish(request):

if request.method == 'POST': return redirect('/ViewStylish/')

else:

sty = Stylish\_details.objects.all()

return render(request, 'ViewStylish.html', {'sty':sty})

def User\_CP(request):

if request.method == 'POST':

CurrentPassword = request.POST['CurrentPassword'] NewPassword = request.POST['NewPassword'] ConfirmPassword = request.POST['ConfirmPassword'] uid = request.session['Stylish\_id']

CurrUser = Stylish\_details.objects.all().filter(id=uid) if CurrUser[0].Password == CurrentPassword:

if NewPassword == ConfirmPassword: Stylish\_details.objects.filter(id=uid).update(Password=NewPassword) messages.info(request,'Passwords Changed Successfully')

return render(request, 'User\_CP.html', {}) else:

messages.info(request,'New Passwords doesnt match') return render(request, 'User\_CP.html', {})

else:

messages.info(request,'Current Password doesnt match') return render(request, 'User\_CP.html', {})

else:

return render(request, 'User\_CP.html', {}) def ChangePassword(request):

if request.method == 'POST':

CurrentPassword = request.POST['CurrentPassword'] NewPassword = request.POST['NewPassword'] ConfirmPassword = request.POST['ConfirmPassword'] uid = request.session['Stylish\_id']

CurrUser = Stylish\_details.objects.all().filter(id=uid) if CurrUser[0].Password == CurrentPassword:

if NewPassword == ConfirmPassword: Stylish\_details.objects.filter(id=uid).update(Password=NewPassword) messages.info(request,'Passwords Changed Successfully')

return render(request, 'ChangePassword.html', {}) else:

messages.info(request,'New Passwords doesnt match') return render(request, 'ChangePassword.html', {})

else:

messages.info(request,'Current Password doesnt match') return render(request, 'ChangePassword.html', {})

else:

return render(request, 'ChangePassword.html', {}) def AcceptStylish(request):

if request.method == 'POST': return redirect('/AcceptStylish/')

else:

sty = Stylish\_details.objects.all().filter(Status= 'Pending') return render(request, 'AcceptStylish.html', {'sty':sty})

def ManageSubscription(request): if request.method == 'POST':

return redirect('/ManageSubscription/') else:

sty = Subscriptions.objects.all().filter(id='1') half = sty[0].halfyear

print('half',half) year = sty[0].year

print('year',year)

return render(request, 'ManageSubscription.html', {'half':half,'year':year}) def AcceptRequest(request,id):

Stylish\_details.objects.filter(id=id).update(Status='Confirmed') messages.info(request,'Requested Accepted')

return redirect('/AcceptStylish/') def UpdateSubs(request):

if request.method == 'POST': half = request.POST['half'] year = request.POST['year']

Subscriptions.objects.filter(id='1').update(halfyear=half,year=year) messages.info(request,'Pricing Changed')

return redirect('/ManageSubscription/') else:

return redirect('/ManageSubscription/') def Chatreply(request):

MType = request.POST.get('MessageType') x = datetime.datetime.now()

currdate = x.strftime('%Y-%m-%d %H:%M:%S') if MType == "Text":

inputtext = request.POST.get('text') Sid = request.POST.get('Sid')

chat = Chat\_details( Type = MType,MsgFrom = "User",Context = inputtext,ContextImage = '',Uid = request.session['User\_id'],Sid = Sid,Datetime=currdate)

chat.save() else:

print("enter Image")

inputtext =request.POST.get('image') print("inputt IMage",inputtext)

Sid = request.POST.get('Sid')

chat = Chat\_details( Type = MType,MsgFrom = "User",Context = '',ContextImage = inputtext,Uid = request.session['User\_id'],Sid = Sid,Datetime=currdate)

chat.save() answer = "Success" data = {

'respond': answer

}

return JsonResponse(data) def StylishChatreply(request):

MType = request.POST.get('MessageType') x = datetime.datetime.now()

currdate = x.strftime('%Y-%m-%d %H:%M:%S') if MType == "Text":

inputtext = request.POST.get('text') Uid = request.POST.get('Uid')

chat = Chat\_details( Type = MType,MsgFrom = "Stylish",Context = inputtext,ContextImage = '',Uid = Uid ,Sid = request.session['Stylish\_id'],Datetime=currdate)

chat.save() else:

print("enter Image")

inputtext =request.POST.get('image') print("inputt IMage",inputtext)

Uid = request.POST.get('Uid')

chat = Chat\_details( Type = MType,MsgFrom = "Stylish",Context = '',ContextImage = inputtext,Uid = Uid ,Sid = request.session['Stylish\_id'],Datetime=currdate)

chat.save() answer = "Success" data = {

'respond': answer

}

return JsonResponse(data) def FillUserChat(request):

Sid = request.POST.get('Sid') UserId = request.session['User\_id'] answer = ""

chattype = "" chatcontent = ""

chat = Chat\_details.objects.filter(Uid=UserId,Sid=Sid).order\_by('Datetime') for x in chat:

chattype = x.Type

if chattype == "Text":

chatcontent = x.Context else:

chatcontent = x.ContextImage chatmessagefrom = x.MsgFrom chatdatetime = x.Datetime

answer +=

chattype+"&"+str(chatcontent)+"&"+chatmessagefrom+"&"+str(chatdatetime)+ "#"

#print(answer) data = {

'respond': answer

}

return JsonResponse(data) def FillStylishChat(request):

UserId = request.POST.get('Uid') Sid = request.session['Stylish\_id'] answer = ""

chattype = "" chatcontent = ""

chat = Chat\_details.objects.filter(Uid=UserId,Sid=Sid).order\_by('Datetime') for x in chat:

chattype = x.Type

if chattype == "Text": chatcontent = x.Context

else:

chatcontent = x.ContextImage chatmessagefrom = x.MsgFrom chatdatetime = x.Datetime

answer +=

chattype+"&"+str(chatcontent)+"&"+chatmessagefrom+"&"+str(chatdatetime)+ "#"

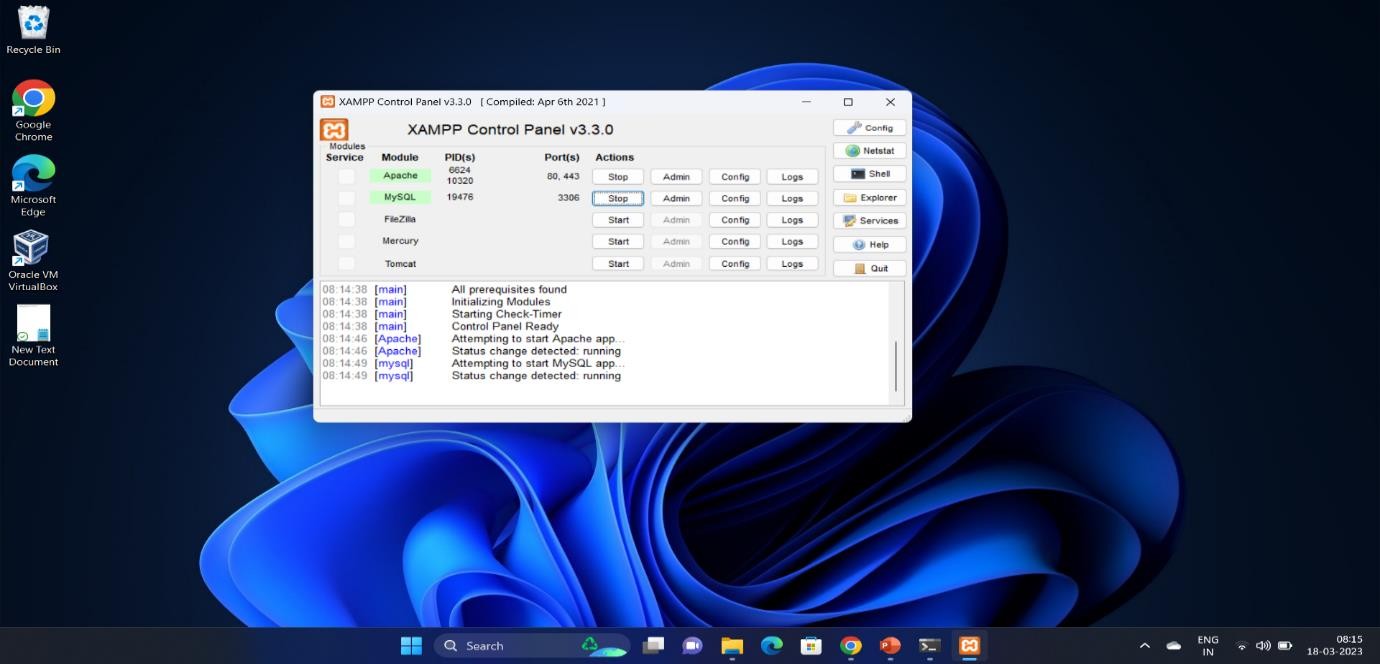
#print(answer) data = {

'respond': answer

}

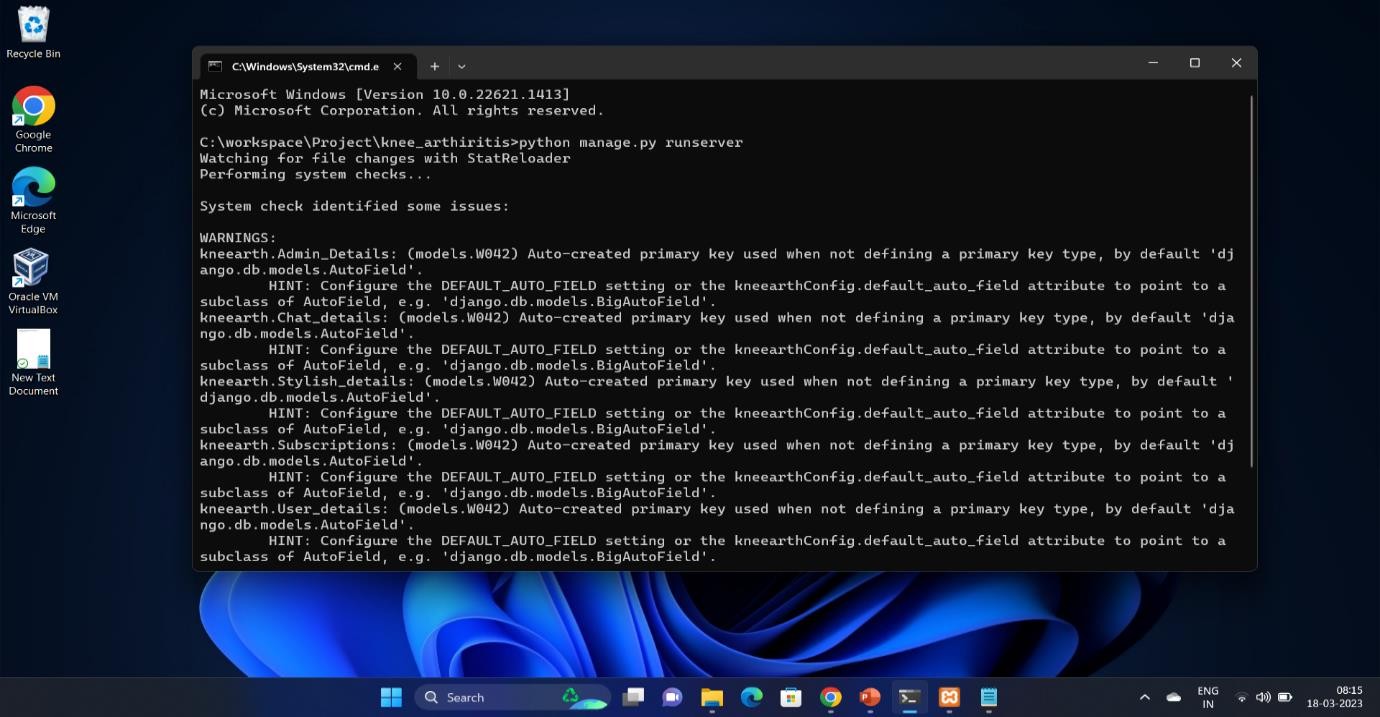
return JsonResponse(data)

* 1. **SNAPSHOTS**



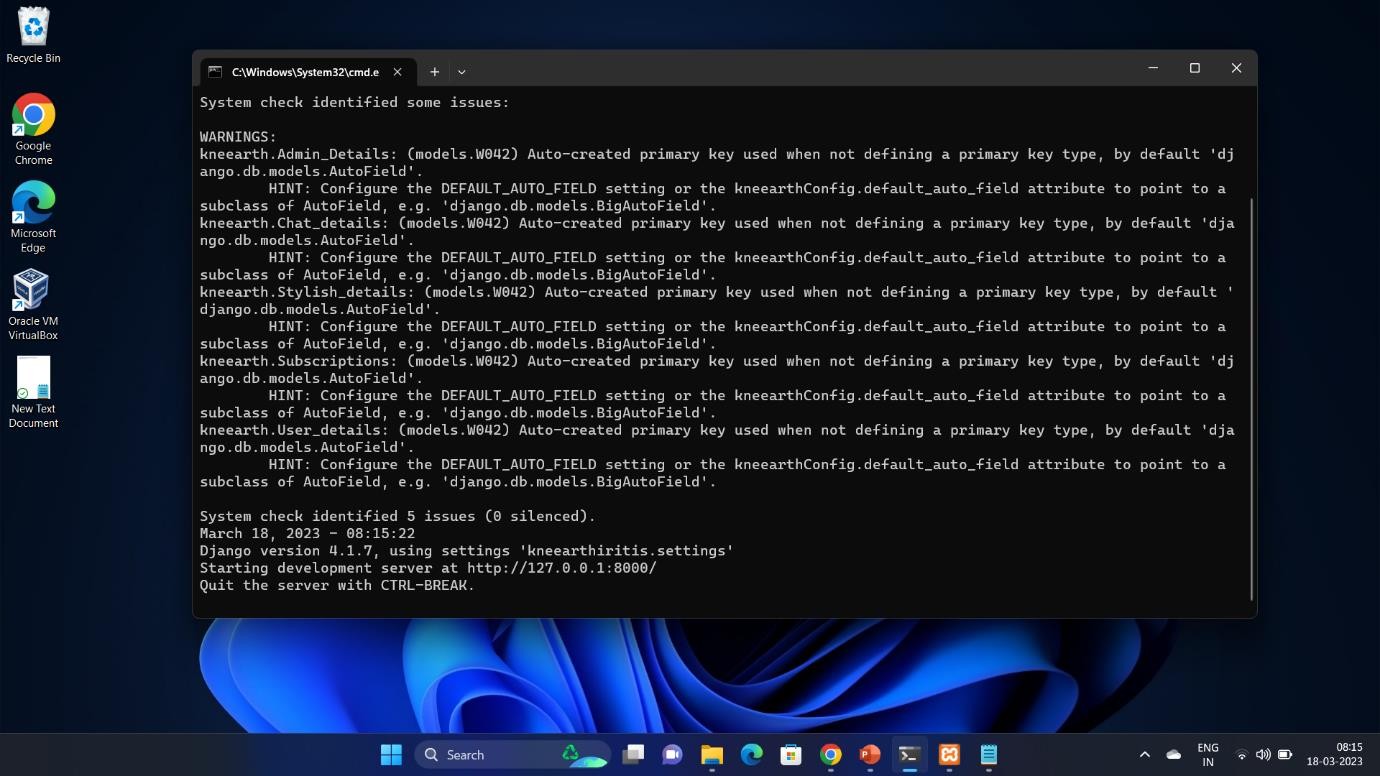
* + 1. **XAMPP Control Panel**

XAMPP control panel is used to start the Apache server and the SQL database. It is the first step to start the both the Apache and SQL to run the project.



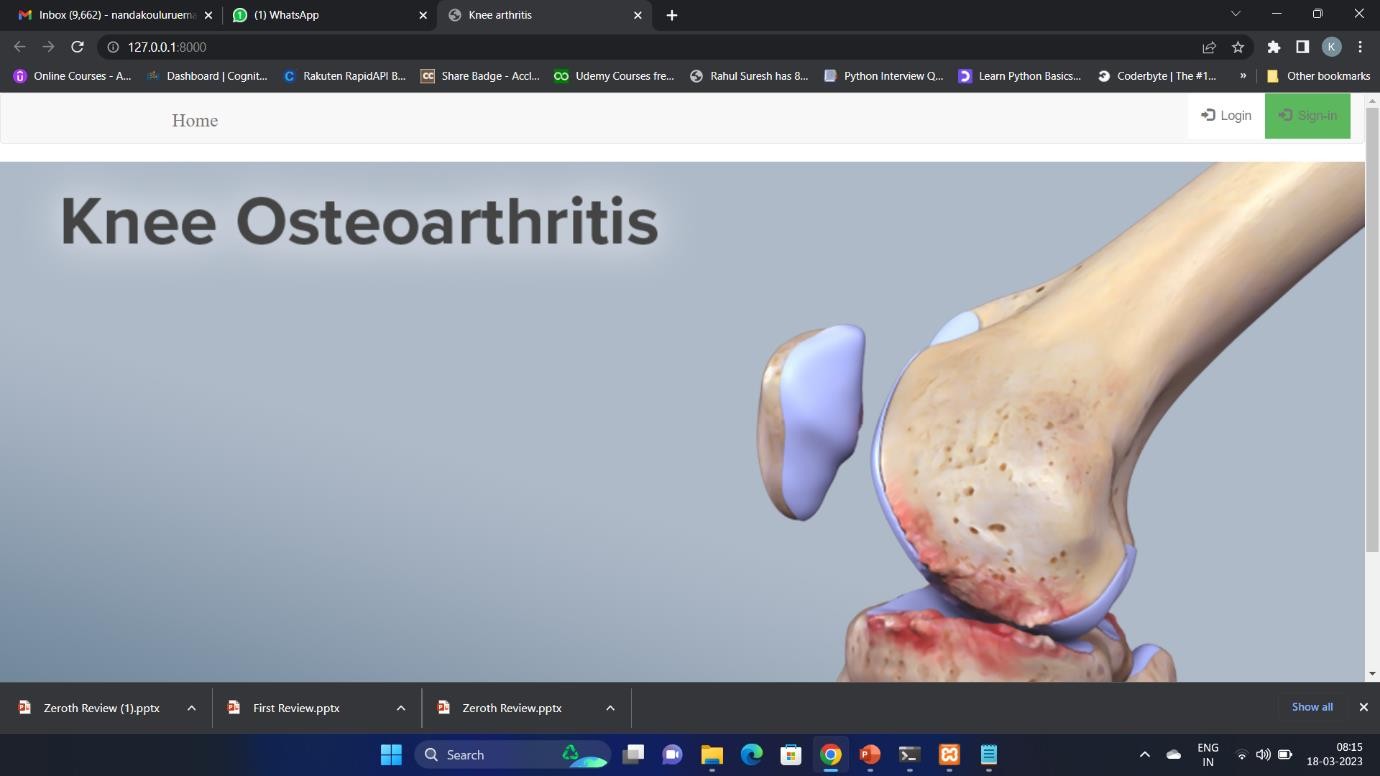
* + 1. **Python code in cmd**

The project will be run by using the code “python manage.py runserver” In the Command prompt of the python.



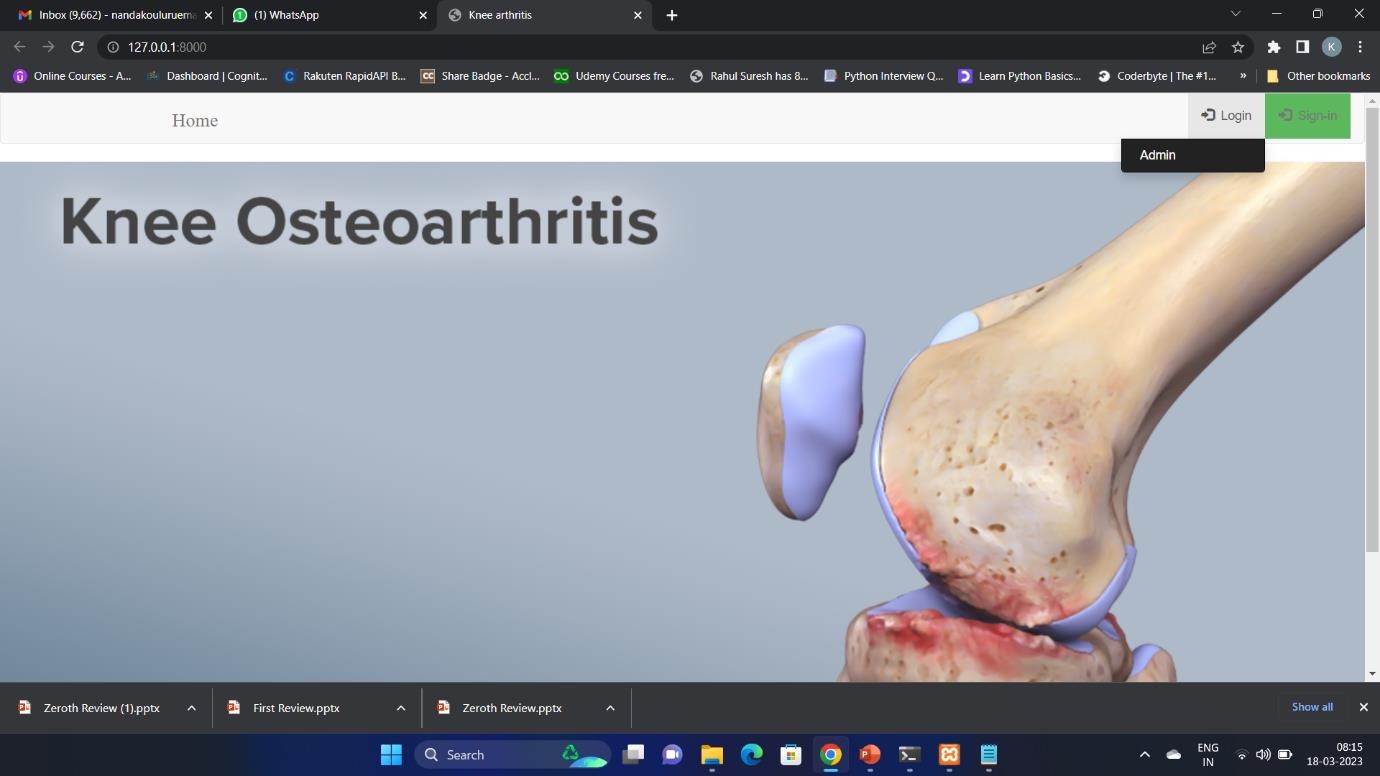
* + 1. **Local Host link in cmd**

After running the code at the end link will be generated to use in the local host browser.



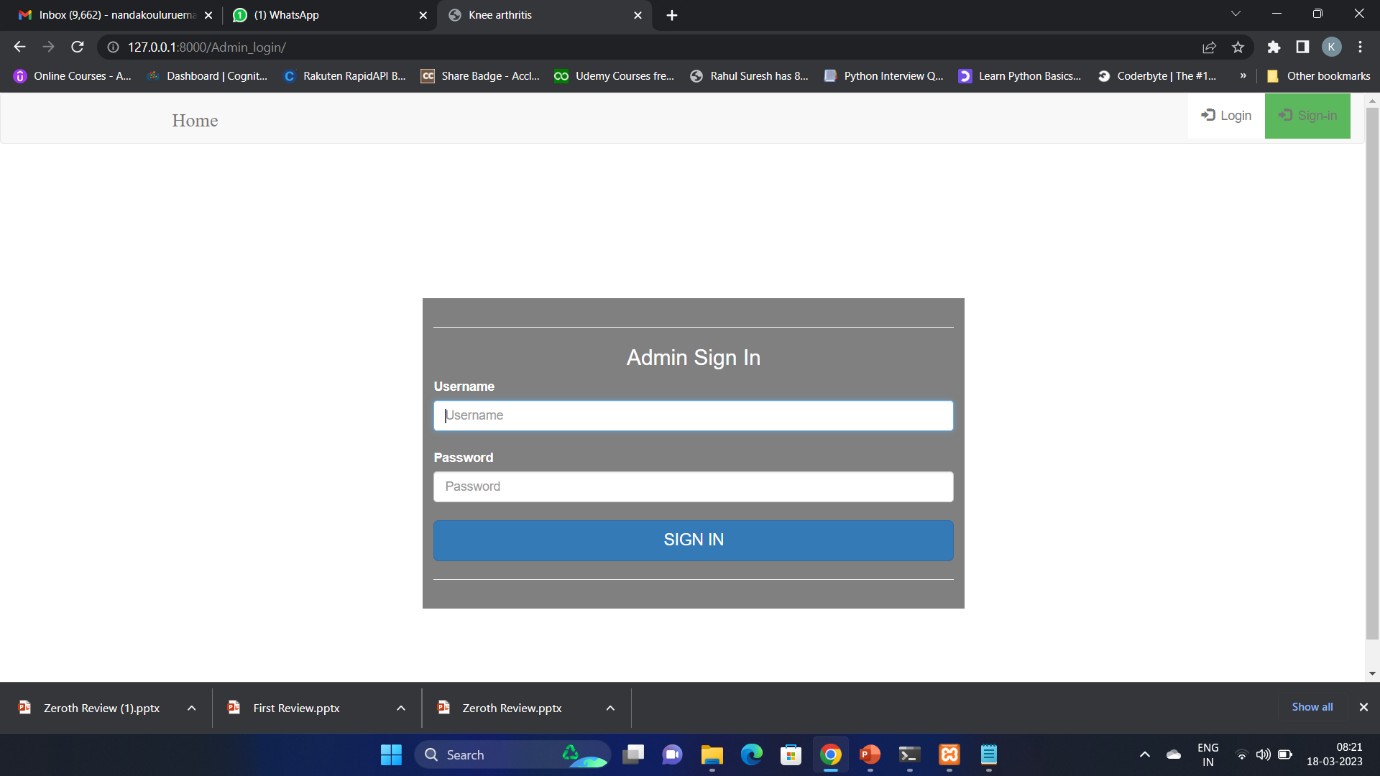
* + 1. **Home Page of the project**

The local host link will open this project webpage. It will contain the both the login and sign-in buttons for the Admin and patient.



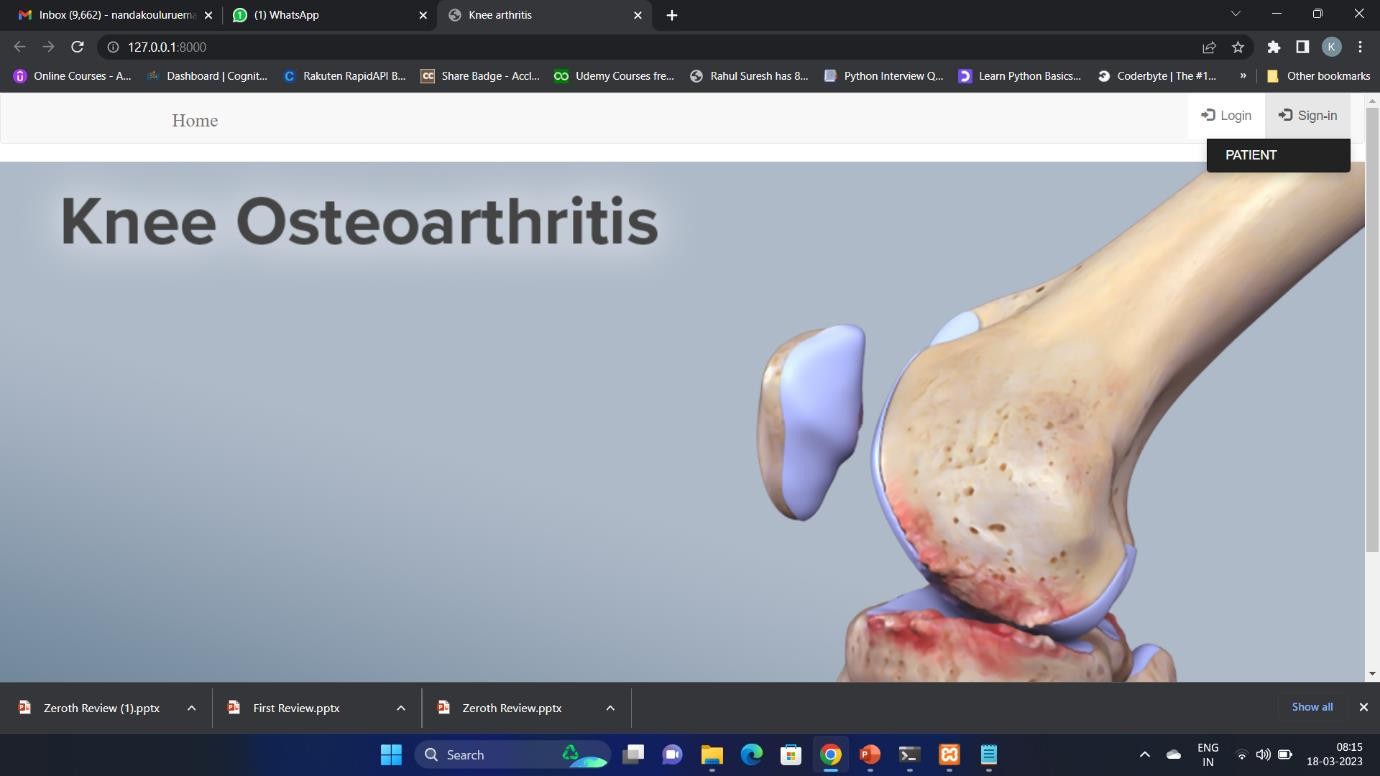
* + 1. **Login Button for Admin**

The above picture indicates about the login button for admin. That will give the access to move into the login page for the Admin



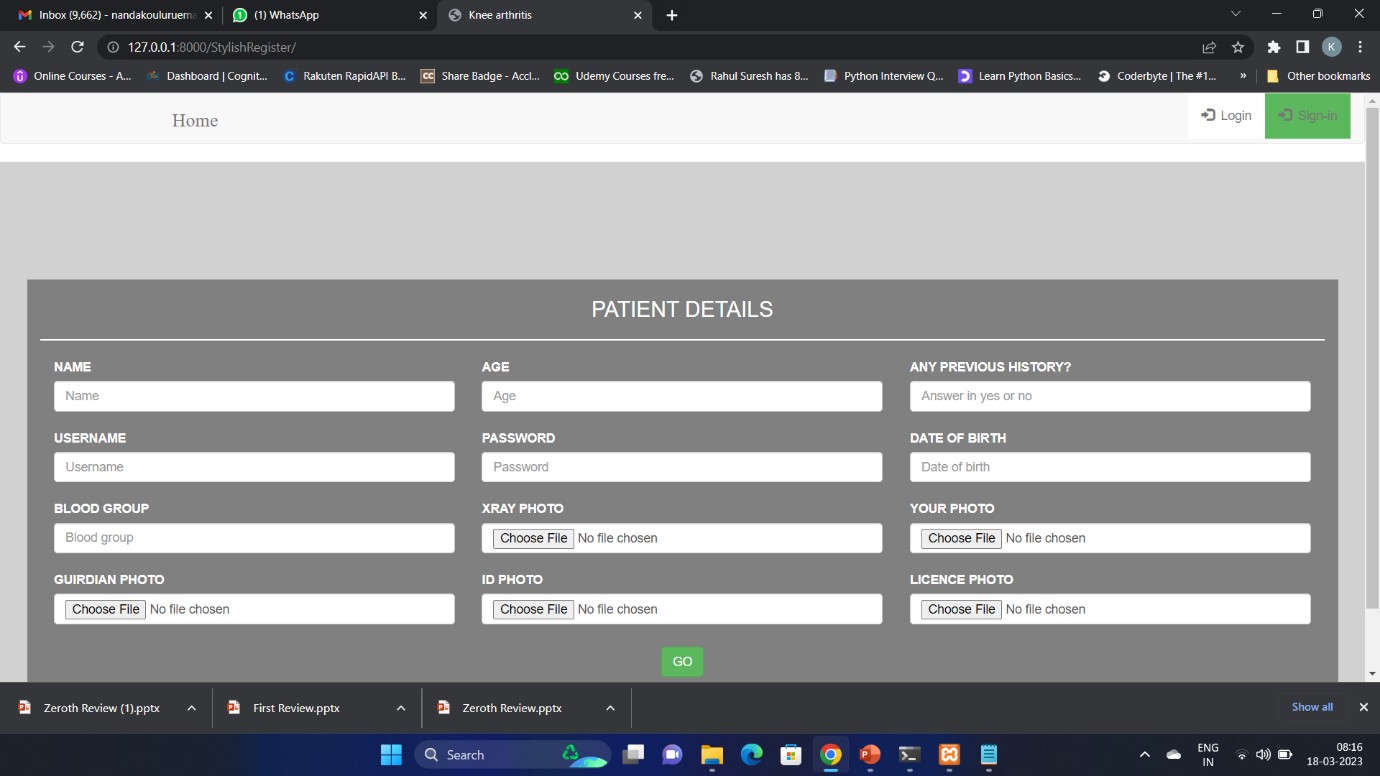
* + 1. **Admin Sign In page**

After Clicking on the login the Admin sign in page will be available. There Admin need to enter the username and the password for sign in.



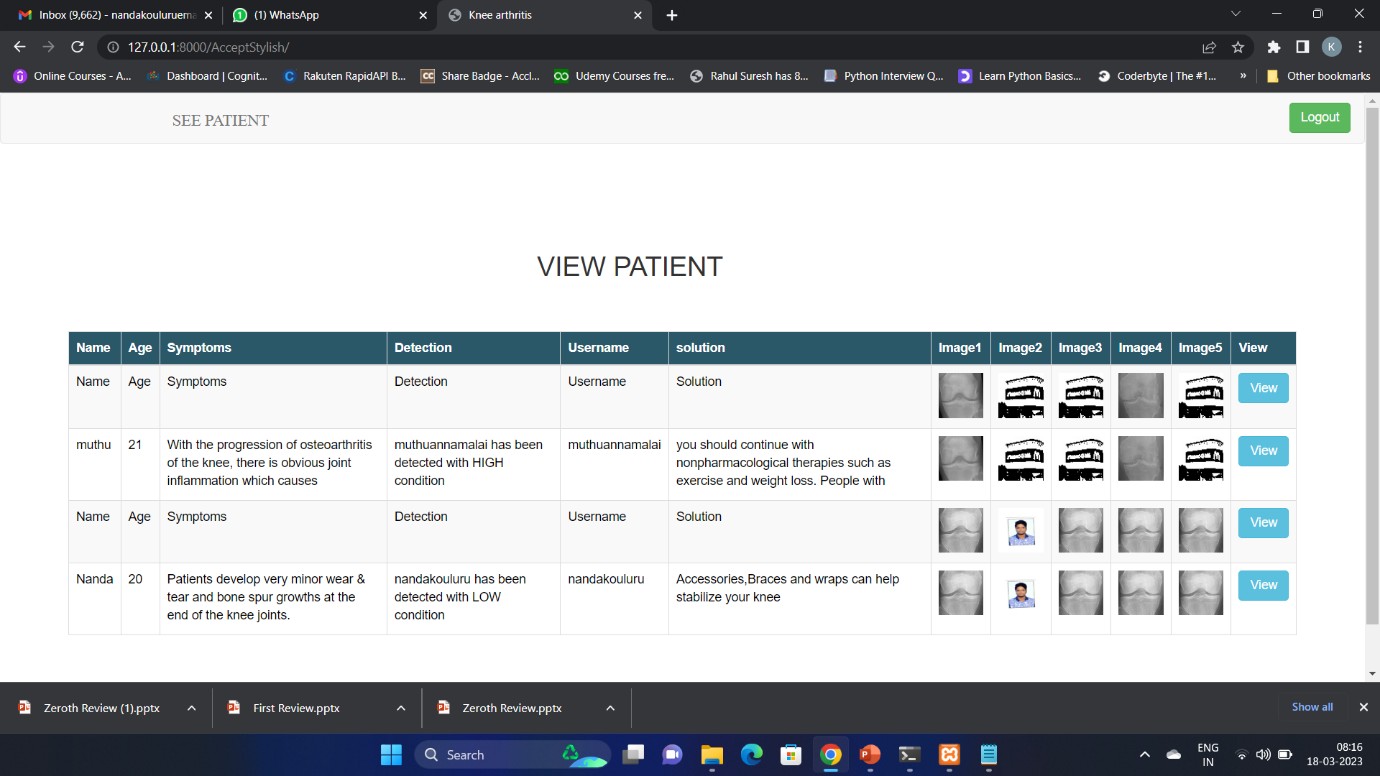
* + 1. **Patient Button for Sign In**

The above picture indicates about the patient button for sign in. It will give the access to move into the page where they need to enter the details.



* + 1. **Patient Details**

The patient need to enter the every details like it should be the Name ,Age, Username ,Password ,DOB ,x-ray photos etc. The username will be unique.



* + 1. **Output for the patient Problem**

The patient details enter will undergo into the database and it will checked by the code and it will release the symptoms ,Detection and the solution.

**REFERENCES**

1. Long, Huibin, et al. “Burden of osteoarthritis in China, 1990 – 2017: findings from the Global Burden of Disease Study 2017,” The Lancet Rheumatology, vol. 2, no. 3, pp. e164- e172, 2020.
2. Li, M. H., et al. “Regenerative approaches for cartilage repair in the treatment of osteoarthritis,” Osteoarthritis and cartilage, vol. 25, no. 10, pp. 1577-1587, 2017.
3. Kellgren, Jonas H., and JS1006995 Lawrence. “Radiological assessment of osteo-arthrosis,” Annals of the rheumatic diseases, vol. 16, no. 4, pp. 494, 1957.
4. Li, Zhenliang, et al. “Deep Multi-Instance Learning with Induced Self- Attention for Medical Image Classification,” 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2020.
5. Su, Ying, Dan Li, and Xiaodong Chen. “Lung nodule detection based on faster R-CNN framework,” Computer Methods and Programs in Biomedicine, vol. 200, pp. 105866, 2021.
6. Liu, Ning, Liangliang Liu, and Jianxin Wang. “Local Adaptive U-net for Medical Image Segmentation,” 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, 2020.
7. Abedin, J., Antony, J., McGuinness, K., Moran, K., O’Connor, N., Rebholz-Schuhman, D., Newell, J.: Predicting knee osteoarthritis severity: comparative modeling based on patient’s data and plain x-ray images. Scientific Reports 9 (04 2019)
8. Antony, J., McGuinness, K., Moran, K., O’Connor, N.E.: Automatic detection of knee joints and quantification of knee osteoarthritis severity using convolutional neural networks. In: Perner, P. (ed.) Machine Learning and Data Mining in Pattern Recognition. pp. 376–390. Springer International Publishing, Cham (2017)
9. Bayramoglu, N., Nieminen, M.T., Saarakkala, S.: A lightweight CNN and joint shape-joint space (js2) descriptor for radiological osteoarthritis detection. In: Medical Image Understanding and Analysis - 24th Annual Conference, MIUA 2020, Oxford, UK, July 15-17, 2020, Proceedings. vol. 1248. Springer (2020)
10. Chen, P., Gao, L., Shi, X., Allen, K., Yang, L.: Fully automatic knee osteoarthritis severity grading using deep neural networks with a novel ordinal loss. Computerized medical imaging and graphics : the official journal of the Computerized Medical Imaging Society 75, 84–92 (2019)
11. Kour, N., Gupta, S., Arora, S.: A survey of knee osteoarthritis assessment based on gait. Archives of Computational Methods in Engineering 28 (01 2020)
12. Kumar, A., Saxena, P.: Quantification of cartilage loss for automatic detection and classification of osteoarthritis using machine learning approach. In: 2019 10th International

Conference on Computing, Communication and Networking Technologies (ICCCNT). pp. 1– 6 (2019)

1. Liu, B., Luo, J., Huang, H.: Toward automatic quantification of knee osteoarthritis severity using improved faster r-cnn. International Journal of Computer Assisted Radiology and Surgery 15 (01 2020)
2. Mehmood, A., Maqsood, M., Bashir, M., Shuyuan, Y.: A deep siamese convolution neural network for multi-class classification of alzheimer disease. Brain Sciences 10(2) (2020)
3. Tiulpin, A., Thevenot, J., Rahtu, E., Lehenkari, P., Saarakkala, S.: Automatic knee osteoarthritis diagnosis from plain radiographs: A deep learning-based approach. Scientific Reports 8 (01 2018)
4. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradientbased localization. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 618–626 (2017)
5. Saleem, M., Farid, M.S., Saleem, S., Khan, M.: X-ray image analysis for automated knee osteoarthritis detection. Signal, Image and Video Processing 14 (09 2020).
6. Mulyanto, M., Faisal, M., Prakosa, S.W., Leu, J.S.: Effectiveness of focal loss for minority classification in network intrusion detection systems. Symmetry 13(1) (2021)