**A Mini Project Report on**

**PREDICTING RISK OF HEART ATTACKS**

#### Submitted to

**Jawaharlal Nehru Technological University, Hyderabad**

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

**BACHELOR OF TECHNOLOGY**

### in

**COMPUTER SCIENCE AND ENGINEERING**

**by**

|  |  |
| --- | --- |
| **Kotapati Venkata Swaraj** | **(18BD1A050R)** |
| **Kotla Venkata Sasank** | **(18BD1A050T)** |
| **Koutarapu Vamsi Mohan** | **(18BD1A050U)** |
| **Vivek Rupender Gopu** | **(18BD1A051N)** |

#### Under the guidance of

#### MR. VENKATESWARLU G

Assistant Professor

Department of CSE



**Department of Computer Science and Engineering**

**KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY**

### Approved by AICTE, Affiliated to JNTUH

### 3-5-1206, Narayanaguda, Hyderabad, TS- 500029

2020 – 2021

 **KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY**

#### *(Accredited by NBA & NAAC, Approved by A.I.C.T.E.,*

***Reg by Govt of Telangana State & Affiliated to JNTU, Hyderabad)***

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CERTIFICATE**

This is to certify that the project entitled **PREDICTING RISK OF HEART ATTACKS** being submitted by

|  |  |
| --- | --- |
| **Kotapati Venkata Swaraj** | **(18BD1A050R)** |
| **Kotla Venkata Sasank** | **(18BD1A050T)** |
| **Koutarapu Vamsi Mohan** | **(18BD1A050U)** |
| **Vivek Rupender Gopu** | **(18BD1A051N)** |

in partial fulfilment for the award of **Bachelor of Technology** in **Computer Science and Engineering** affiliated to the **Jawaharlal Nehru Technological University, Hyderabad** during the year 2021-22**.**

Internal Guide Head of the Department

**(MR. VENKATESWARALU G) (DR. S. PADMAJA)**

**Submitted for Viva Voce Examination held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**External Examiner**

***Unit of Keshav Memorial Educational Society***

#: 3-5-1026 Narayanaguda Hyderabad 500029.

 040-3261407  [www.kmit.in](http://www.kmit.in/) e-mail: [principal@kmit.in](mailto:principal@kmit.in)

**Vision of KMIT**

Producing quality graduates trained in the latest technologies and related tools and striving to make India a world leader in software and hardware products and services. To achieve academic excellence by imparting in depth knowledge to the students, facilitating research activities and catering to the fast growing and ever- changing industrial demands and societal needs.

**Mission of KMIT**

* To provide a learning environment that helps students to enhance problem solving skills, be successful in their professional lives and to prepare students to be lifelong learners through multi model platforms and educating them about their professional, and ethical responsibilities
* To establish Industry Institute Interaction to make students ready for the industry.
* To provide exposure to students to the latest tools and technologies hardware and software. the area of hardware and software
* To promote research-based projects/activities in the emerging areas of technology convergence.
* To encourage and enable students to not merely seek jobs from the industry but also to create new enterprises.
* To induce in the students a spirit of nationalism which will enable the student to develop and understand India's problems and to encourage them to come up with effective solutions for the same.
* To support the faculty in their endeavors to accelerate their learning curve in order to continue to deliver excellent service to students.

#### Vision & Mission of CSE

**Vision of the CSE**

To be among the region's premier teaching and research Computer Science and Engineering departments producing globally competent and socially responsible graduates in the most conducive academic environment.

#### Mission of the CSE

* To provide faculty with state-of-the-art facilities for continuous professional development and research, both in foundational aspects and of relevance to emerging computing trends.
* To impart skills that transform students to develop technical solutions for societal needs and inculcate entrepreneurial talents.
* To inculcate an ability in students to pursue the advancement of knowledge in various specializations of Computer Science and Engineering and make them industry-ready.
* To engage in collaborative research with academia and industry and generate adequate resources for research activities for seamless transfer of knowledge resulting in sponsored projects and consultancy.
* To cultivate responsibility through sharing of knowledge and innovative computing solutions that benefits the society-at-large.
* To collaborate with academia, industry and community to set high standards in academic excellence and in fulfilling societal responsibilities.

#### PROGRAM OUTCOMES (POs)

1. **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problem and design system component or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create select, and, apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to societal, health, safety. Legal und cultural issues and the consequent responsibilities relevant to professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

#### PROGRAM SPECIFIC OUTCOMES (PSOs)

**PSO1:** An ability to analyze the common business functions to design and develop appropriate Information Technology solutions for social upliftment.

**PSO2:** Shall have expertise on the evolving technologies like Mobile Apps, CRM, ERP, Big Data, etc.

#### PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

**PEO1:** Graduates will have successful careers in computer related engineering fields or will be able to successfully pursue advanced higher education degrees.

**PEO2:** Graduates will try and provide solutions to challenging problems in their profession by applying computer engineering principles.

**PEO3:** Graduates will engage in life-long learning and professional development by rapidly adapting changing work environment.

**PEO4:** Graduates will communicate effectively, work collaboratively and exhibit high levels of professionalism and ethical responsibility.

**PROJECT OUTCOMES**

**P1:** Students register their complaints and get them resolved by higher authorities.

**P2:** It helps college authorities to learn the dissatisfaction of their students.

**P3:** Admin/Cell member further forward the complaints to required departments.

**P4:** College authorities understand students’ grievance and resolve them.

**LOW – 1**

**MEDIUM – 2**

**HIGH - 3**

#### PROJECT OUTCOMES MAPPING PROGRAM OUTCOMES

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PO** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** |
| **P1** | M | M |  | H |  |  | M |  | M | M | M | M |
| **P2** |  | M |  | H |  | M | M |  | M | M |  | L |
| **P3** | M | M |  | H |  |  | M |  |  |  | M |  |
| **P4** | M | M |  | H |  | M | M |  | M |  |  | M |

**PROJECT OUTCOMES MAPPING PROGRAM SPECIFIC OUTCOMES**

|  |  |  |
| --- | --- | --- |
| **PSO** | **PSO1** | **PSO2** |
| **P1** | H | M |
| **P2** | H | M |
| **P3** | M | M |
| **P4** | M | M |

**PROJECT OUTCOMES MAPPING PROGRAM EDUCATIONAL OBJECTIVES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PEO** | **PEO1** | **PEO2** | **PEO3** | **PEO4** |
| **P1** |  | M | L | L |
| **P2** | L | M |  | M |
| **P3** |  | M |  | L |
| **P4** | L | M |  | M |

## DECLARATION

We hereby declare that the project report entitled **“PREDICTING RISK OF HEART ATTACKS”** is done in the partial fulfillment for the award of the Degree in Bachelor of Technology in Computer Science and Engineering affiliated to Jawaharlal Nehru Technological University, Hyderabad. This project has not been submitted anywhere else.

#### Kotapati Venkata Swaraj (18BD1A050R)

**Kotla Venkata Sasank (18BD1A050T)**

#### Koutarapu Vamsi Mohan (18BD1A050U)

**Vivek Rupender Gopu (18BD1A051N)**

## ACKNOWLEDGMENT

We take this opportunity to thank all the people who have rendered their full support to our project work.

We render our thanks to **Dr. Maheshwar Dutta**, B.E., M Tech., Ph.D., Principal who encouraged us to do the Project.

We are grateful to **Mr. Neil Gogte**, Director for facilitating all the amenities required for carrying out this project.

We express our sincere gratitude to **Mr. S. Nitin**, Director and **Mrs. Anuradha**, Dean Academics for providing an excellent environment in the college.

We are also thankful to **Dr. S. Padmaja**, Head of the Department for providing us with both time and amenities to make this project a success within the given schedule.

We are also thankful to our guide **Mr. Venkateswarlu G** for his valuable guidance and encouragement given to us throughout the project work.

We would like to thank the entire CSE Department faculty, who helped us directly and indirectly in the completion of the project. We sincerely thank our friends and family for their constant motivation during the project work.

**Kotapati Venkata Swaraj (18BD1A050R)**

**Kotla Venkata Sasank (18BD1A050T)**

**Koutarapu Vamsi Mohan (18BD1A050U)**

**Vivek Rupender Gopu (18BD1A051N)**

## CONTENTS

[ABSTRACT i](#_bookmark0)

[LIST OF FIGURES ii](#_bookmark1)

[LIST OF TABLES iii](#_bookmark2)

1. [INTRODUCTION 2](#_bookmark3)
   1. [Purpose of project 2](#_bookmark4)
   2. [Problems with existing system 2](#_bookmark5)
   3. [Proposed system 2](#_bookmark6)
   4. [Scope of the project 3](#_bookmark7)
2. [SYSTEM REQUIREMENTS SPECIFICATION 5](#_bookmark8)
   1. [What is SRS ? 5](#_bookmark9)
   2. [Role of SRS 5](#_bookmark10)
   3. [Requirements Specification Document 5](#_bookmark11)
   4. [Functional Requirements 6](#_bookmark12)
   5. [Non Functional Requirements 7](#_bookmark13)
   6. [Software Requirements 7](#_bookmark14)
   7. [Hardware Requirements 8](#_bookmark15)
3. [LITERATURE SURVEY 10](#_bookmark16)
4. [SYSTEM DESIGN 16](#_bookmark17)
   1. [Introduction to UML 16](#_bookmark18)
   2. [UML Diagram 17](#_bookmark19)
      1. [Use case diagram 17](#_bookmark20)
      2. [Class Diagram 22](#_bookmark21)
      3. [Sequence diagram 24](#_bookmark22)
5. [IMPLEMENTATION 27](#_bookmark23)
   1. [Code Snippets 27](#_bookmark24)
6. [TESTING 37](#_bookmark25)
   1. [Introduction to Testing 37](#_bookmark26)
      1. [Functional Testing 37](#_bookmark27)
      2. [Software Testing Life Cycle 38](#_bookmark28)
   2. [Test Cases 39](#_bookmark29)
      1. [Student Test Cases 39](#_bookmark30)
      2. [Admin Test Cases 39](#_bookmark31)
      3. [Cell Member Test Cases 39](#_bookmark32)
7. [SCREENSHOTS 45](#_bookmark33)
8. [FUTURE ENHANCEMENTS 53](#_bookmark34)
9. [CONCLUSION 55](#_bookmark35)
10. [REFERENCES 57](#_bookmark36)

## ABSTRACT

Cardiovascular diseases kill approximately 17 million people globally every year, and they mainly exhibit myocardial infarctions and heart failures. Heart failure (HF) occurs when the heart cannot pump enough blood to meet the needs of the body. Available electronic medical records of patients quantify symptoms, body features, and clinical laboratory test values, which can be used to perform biostatistics analysis aimed at highlighting patterns and correlations otherwise undetectable by medical doctors.

Machine learning, in particular, can predict patients’ survival from their data and can individuate the most important features among those included in their medical records. This discovery has the potential to impact clinical practice, becoming a new supporting tool for physicians when predicting if a heart failure patient will survive or not. Indeed, medical doctors aiming at understanding if a patient will survive after heart failure may focus mainly on serum creatinine and ejection fraction.

## LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **S.NO** | **LIST OF FIGURES** | **PAGE NO** |
| 4.2.1.1 | Use case diagram for Student | 18 |
| 4.2.1.2 | Use case diagram for Admin | 19 |
| 4.2.1.3 | Use case diagram for Cell member | 20 |
| 4.2.1.4 | Use case diagram | 21 |
| 4.2.2.1 | Class diagram | 23 |
| 4.2.3.1 | Sequence diagram, | 25 |
| 7.1 | Home page | 44 |
| 7.2 | Student registration | 44 |
| 7.3 | Student login | 45 |
| 7.4 | Student dashboard | 45 |
| 7.5 | Complaints history | 46 |
| 7.6 | Register complaint | 46 |
| 7.7 | Admin login | 47 |
| 7.8 | Admin dashboard | 47 |
| 7.9 | Add Cell member | 48 |
| 7.10 | In process complaints (Admin) | 48 |
| 7.11 | Complaint details | 49 |
| 7.12 | Complaint report | 49 |
| 7.13 | Cell member dashboard | 50 |
| 7.14 | Not yet processed complaints (Cell member) | 50 |

## LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **S. NO** | **LIST OF TABLES** | **PAGE NO** |
| 6.2.1 | Test case 1 | 39 |
| 6.2.2 | Test case 2 | 39 |
| 6.2.3 | Test case 3 | 40 |
| 6.2.4 | Test case 4 | 40 |
| 6.2.5 | Test case 5 | 41 |
| 6.2.6 | Test case 6 | 41 |
| 6.2.7 | Test case 7 | 42 |

# CHAPTER -1

## INTRODUCTION

### Purpose of project

Cardiovascular diseases (CVDs) are the leading cause of death globally. An estimated 17.9 million people died from CVDs in 2019, representing 32% of all global deaths. Of these deaths, 85% were due to heart attack and stroke. In particular, heart failure occurs when the heart is unable to pump enough blood to the body, and it is usually caused by diabetes, high blood pressure, or other heart conditions or diseases. Medical doctors and physicians have failed to accurately forecast heart failure-related events in clinical practice. Heart disease has created a lot of serious concern among researchers. One of the major challenges in heart disease is correct detection and finding the presence of it inside a human. Early techniques have not been so efficient in finding it, even medical professors are not so efficient enough in predicting heart disease.

In this context, electronic health records (EHRs, also called medical records) can be considered a useful resource of information to unveil hidden and non-obvious correlations and relationships between patients’ data, not only for research but also for clinical practice and to better understand the unknown risk factors. Machine learning applied to medical records, in particular, can be an effective tool both to predict the survival of each patient having heart failure symptoms. It can be used to detect the most important clinical features (or risk factors) that may lead to heart failure.

### Problems with existing system

There are various medical instruments available in the market for predicting heart disease. There are two major problems in them, the first one is that they are very expensive and the second one is that they are not efficiently able to calculate the chance of heart disease in humans. According

to the latest survey conducted by WHO, the medical profession was able to correctly predict only 67% of heart disease. So, there is a vast scope of research in the area of predicting heart disease in humans.

### Proposed system

The main task is to develop a model that accurately predicts failure of heart. Such models can be trained using classification algorithms, some of the ML classification algorithms are Logistic Regression, Support Vector Machine, Decision Tree Classifier, Gradient Booster Classifier, etc. Dataset contains the medical records of 299 heart failure patients, which can be used for training the model. The dataset contains 13 features, which report clinical, body, and lifestyle information. Some features are binary anemia, high blood pressure, diabetes, sex, and smoking. The proposed dataset can be used to predict patients’ survival, by applying any of the different classification methods in machine learning. The classifiers include linear statistical method (Linear Regression), tree-based methods (Random Forests, Decision Tree), Artificial Neural Networks, Support Vector Machines, instance-based learning model (k-Nearest Neighbors), probabilistic classifier (Naïve Bayes), and ensemble boosting method (Gradient Boosting).

### Scope of the project

This Machine Learning Model is used to predict the risk of heart attack in a patient, given vital information like age, anemia, diabetes, ejection fraction, high blood pressure, platelets, serum creatinine, gender, smoking, etc.

# CHAPTER -2

## SYSTEM REQUIREMENTS SPECIFICATION

### What is SRS?

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase.) The SRS phase consists of two basic activities:

#### Problem/Requirement Analysis:

The process is order and more nebulous of the two, deals with understand the problem, the goal and constraints.

#### Requirement Specification:

Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity.

The Requirement phase terminates with the production of the validate SRS document.

Producing the SRS document is the basic goal of this phase.

### Role of SRS

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium though which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

### Requirements Specification Document

A Software Requirements Specification (SRS) is a document that describes the nature of a project, software or application. In simple words, SRS document is a manual of a project

provided it is prepared before you kick-start a project/application. This document is also known by the names SRS report, software document. A software document is primarily prepared for a project, software or any kind of application.

There are a set of guidelines to be followed while preparing the software requirement specification document. This includes the purpose, scope, functional and non-functional requirements, software and hardware requirements of the project. In addition to this, it also contains the information about environmental conditions required, safety and security requirements, software quality attributes of the project etc.

The purpose of SRS (Software Requirement Specification) document is to describe the external behaviour of the application developed or software. It defines the operations, performance and interfaces and quality assurance requirement of the application or software. The complete software requirements for the system are captured by the SRS.

This section introduces the requirement specification document for Predicting Risk of Heart Attacks using Machine Learning which enlists functional as well as non-functional requirements.

### Functional Requirements

For documenting the functional requirements, the set of functionalities supported by the system are to be specified. A function can be specified by identifying the state at which data is to be input to the system, its input data domain, the output domain, and the type of processing to be carried on the input data to obtain the output data. Functional requirements define specific behaviour or function of the application. Following are the functional requirements:

1.

### Non-Functional Requirements

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. Especially these are the constraints the system must work within. Following are the non-functional requirements:

* + 1. Must be able to work on any mobile/laptop.
    2. There should not be any lag in displaying the result.
    3. The model should predict the target feature with high accuracy.
    4. Security of data should not be compromised.
    5. Internet connection is essential for the functioning of the web portal.

#### Performance:

The performance of the developed applications can be calculated by using following methods: Measuring enables you to identify how the performance of your application stands in relation to your defined performance goals and helps you to identify the bottlenecks that affect your application performance. It helps you identify whether your application is moving toward or away from your performance goals. Defining what you will measure, that is, your metrics, and defining the objectives for each metric is a critical part of your testing plan.

Performance objectives include the following:

1. Response time
2. Latency throughput
3. Resource utilization

### Software Requirements

* Operating System : Windows 10, MacOS, Linux
* Platform : Google Colab
* Programming Languages : Python
* External Libraries : Numpy, Pandas, Matplotlib, Scikit-Learn, Seaborn

### Hardware Requirements

* Processor : Intel core i3 or above.
* Hard Disk : 20 GB or above.
* RAM : 2 GB or above.

# CHAPTER - 3

## 3. LITERATURE SURVEY

Prediction of heart disease based on machine learning algorithm is always curious case for researchers recently there is a wave of papers and research material on this area. Contributions of different researchers for the development of this field are:

* Marjia Sultana, Afrin Haider and Mohammad Shorif Uddin have illustrated about how the datasets available for heart disease are generally a raw in nature which is highly redundant and inconsistent. There is a need of pre-processing of these data sets; in this phase high dimensional data set is reduced to low data set. Selection of important features reduces work of training the algorithm and hence resulted in reduction in time complexity.
* Different methods have their own merits and demerits in work done by M.A.Jabbar, B.L Deekshatulu, Priti Chandra , an optimization of feature has been done to achieve higher classification efficiency in Decision Tree . Performance is measured by running algorithms (Bayes Net and SMO) on data set collected from WEKA software and then compared using predictive accuracy, ROC curve, ROC value.
* Different methods have their own merits and demerits in work done by M.A.Jabbar, B.L Deekshatulu, Priti Chandra , an optimization of feature has been done to achieve higher classification efficiency in Decision Tree . It is an approach for early detection of heart disease by utilizing variety of feature. This kind of approach can also be utilized for other spheres of research.
* Yogeswaran Mohan et.al have collected raw data form EEG device and used to train neural network for pattern classification. Authors have got efficiency up to 95% with help of trained neural network watching the success of neural network researches working in the domain of SVM.

### Technologies Used

#### Python Programming Language

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its 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. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a cycle-detecting garbage collection system (in addition to reference counting). Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward-compatible. Python 2 was discontinued with version 2.7.18 in 2020.

#### Numpy

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python.

In Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy. Arrays are very frequently used in data science, where speed and resources are very important.

NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently. This behavior is called locality of reference

in computer science. This is the main reason why NumPy is faster than lists. Also it is optimized to work with latest CPU architectures.

NumPy is a Python library and is written partially in Python, but most of the parts that require fast computation are written in C or C++. The source code for NumPy is located at this github repository https://github.com/numpy/numpy

#### Pandas

Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data. The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.

Pandas allows us to analyze big data and make conclusions based on statistical theories. Pandas can clean messy data sets, and make them readable and relevant. Relevant data is very important in data science.

Pandas are also able to delete rows that are not relevant, or contains wrong values, like empty or NULL values. This is called cleaning the data.

Pandas gives you answers about the data. Like:

* Is there a correlation between two or more columns?
* What is average value?
* Max value?
* Min value?

#### Matplotlib

Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. Matplotlib is written in Python and makes use of NumPy, the numerical mathematics extension of Python. It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPythonotTkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also.

Matplotlib has a procedural interface named the Pylab, which is designed to resemble MATLAB, a proprietary programming language developed by MathWorks. Matplotlib along with NumPy can be considered as the open-source equivalent of MATLAB.

Matplotlib was originally written by John D. Hunter in 2003. The current stable version is 2.2.0 released in January 2018.

Matplotlib is a low level graph plotting library in python that serves as a visualization utility. Matplotlib was created by John D. Hunter. Matplotlib is open source and we can use it freely. Matplotlib is mostly written in python, a few segments are written in C, Objective-C and Javascript for Platform compatibility.

The source code for Matplotlib is located at this github repository https://github.com/matplotlib/matplotlib

#### Scikit-Learn

Scikit-learn (sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon **NumPy, SciPy** and **Matplotlib**.

It was originally called ***scikits.learn*** and was initially developed by David Cournapeau as a Google summer of code project in 2007. Later, in 2010, Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, and Vincent Michel, from FIRCA (French Institute for Research in Computer Science and Automation), took this project at another level and made the first public release (v0.1 beta) on 1st Feb. 2010.

Rather than focusing on loading, manipulating and summarizing data, Scikit-learn library is focused on modeling the data. Some of the most popular groups of models provided by Sklearn are as follows −

**Supervised Learning algorithms** − Almost all the popular supervised learning algorithms, like Linear Regression, Support Vector Machine (SVM), Decision Tree etc., are the part of scikit-learn.

**Unsupervised Learning algorithms** − On the other hand, it also has all the popular unsupervised learning algorithms from clustering, factor analysis, PCA (Principal

Component Analysis) to unsupervised neural networks.

**Clustering** − This model is used for grouping unlabeled data.

**Cross Validation** − It is used to check the accuracy of supervised models on unseen data.

**Ensemble methods** − As name suggest, it is used for combining the predictions of multiple supervised models.

**Dimensionality Reduction** − It is used for reducing the number of attributes in data which can be further used for summarization, visualization and feature selection.

**Feature extraction** − It is used to extract the features from data to define the attributes in image and text data.

**Feature selection** − It is used to identify useful attributes to create supervised models.

#### Seaborn

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of matplotlib library and also closely integrated to the data structures from pandas.

Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs, so that we can switch between different visual representations for same variables for better understanding of dataset.

Plots are basically used for visualizing the relationship between variables. Those variables can be either be completely numerical or a category like a group, class or division. Seaborn divides plot into the below categories –

* Relational plots: This plot is used to understand the relation between two variables.
* Categorical plots: This plot deals with categorical variables and how they can be visualized.
* Distribution plots: This plot is used for examining univariate and bivariate distributions
* Regression plots: The regression plots in seaborn are primarily intended to add a visual guide that helps to emphasize patterns in a dataset during exploratory data analyses.
* Matrix plots: A matrix plot is an array of scatterplots.
* Multi-plot grids: It is a useful approach is to draw multiple instances of the same plot on different subsets of the dataset.

#### HTML5

HTML5 is the latest version of Hypertext Markup Language, the code that describes web pages. It's actually three kinds of code: HTML, which provides the structure; Cascading Style Sheets (CSS), which take care of presentation; and JavaScript, which makes things happen. HTML5 has been designed to deliver almost everything you'd want to do online without requiring additional software such as browser plugins.

It does everything from animation to apps, music to movies, and can also be used to build incredibly complicated applications that run in your browser. We've come a long way since HTML could barely handle a simple page layout. HTML5 can be used to write web applications that still work when you're not connected to the net; to tell websites where you are physically located; to handle high-definition video; and to deliver extraordinary graphics

#### CSS

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable. CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

#### Bootstrap

Bootstrap is the most popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website. It is absolutely free to download and use. It is a frontend framework used for easier and faster web development. It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others. It can also use JavaScript plug-ins. It facilitates you to create responsive designs.

After its opensource release in 2011, Bootstrap became popular very quickly, and not without reason. Web designers and web developers like Bootstrap because it is flexible and easy to work with. Its main advantages are that it is responsive by design, it maintains wide browser compatibility, it offers consistent design by using re-usable components, and it is very easy to use and quick to learn. It offers rich extensibility using JavaScript, coming with built in support for jQuery plugins and a programmatic JavaScript API. Bootstrap can be used with any IDE or editor, and any server-side technology and language, from ASP.NET to PHP to Ruby on Rails.

#### 3.1.10 Flask

#### Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

#### Flask has become popular among Python enthusiasts. As of October 2020, it has second most stars on GitHub among Python web-development frameworks, only slightly behind Django, and was voted the most popular web framework in the Python Developers Survey 2018.

# CHAPTER - 4

## SYSTEM DESIGN

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of system analysis, systems architecture and systems engineering.

### Introduction to UML

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic, semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows:

* + 1. User Model View:

This view represents the system from the users’ perspective. The analysis representation describes a usage scenario from the end-users’ perspective.

* + 1. Structural Model View:

In this model, the data and functionality are arrived from inside the system. This model view models the static structures.

* + 1. Behavioral Model View:

It represents the dynamic of behavioral as parts of the system, depicting he interactions of collection between various structural elements described in the user model and structural model view.

* + 1. Implementation Model View:

In this view, the structural and behavioral as parts of the system are represented as they are to be built.

* + 1. Environmental Model View:

In this view, the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

### UML Diagram

#### Use case diagram

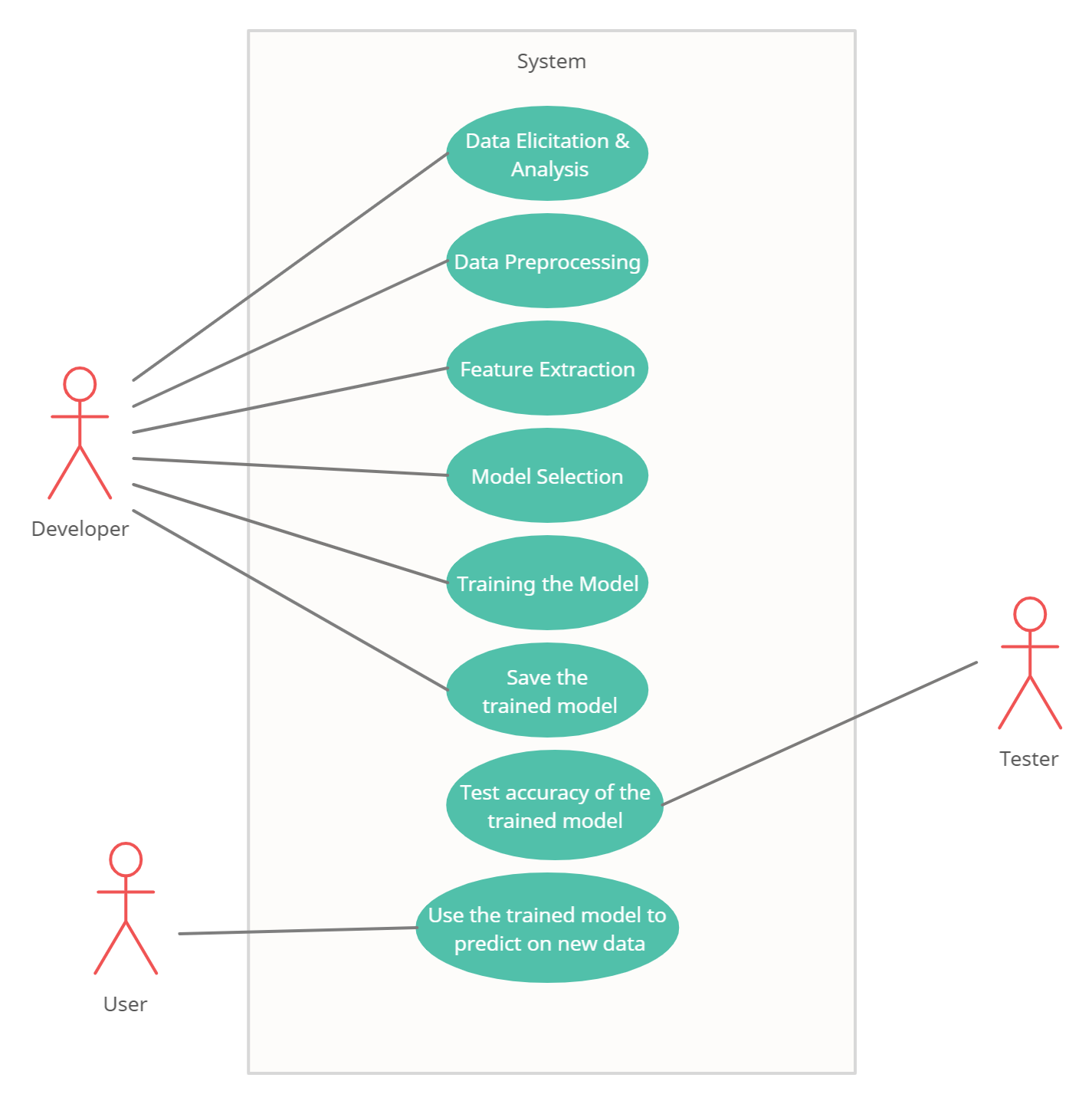
To model a system, the most important aspect is to capture the dynamic behaviour. To clarify a bit in details, dynamic behaviour means the behaviour of the system when it is running/operating.

So only static behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. So use case diagrams are consisting of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system. So to model the entire system numbers of use case diagrams are used.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified. In brief, the purposes of use case diagrams can be as follows:

* + - 1. Used to gather requirements of a system.
      2. Used to get an outside view of a system.
      3. Identify external and internal factors influencing the system.
      4. Show the interacting among the requirements are actors.



#### Fig 4.2.1.1 Use case diagram

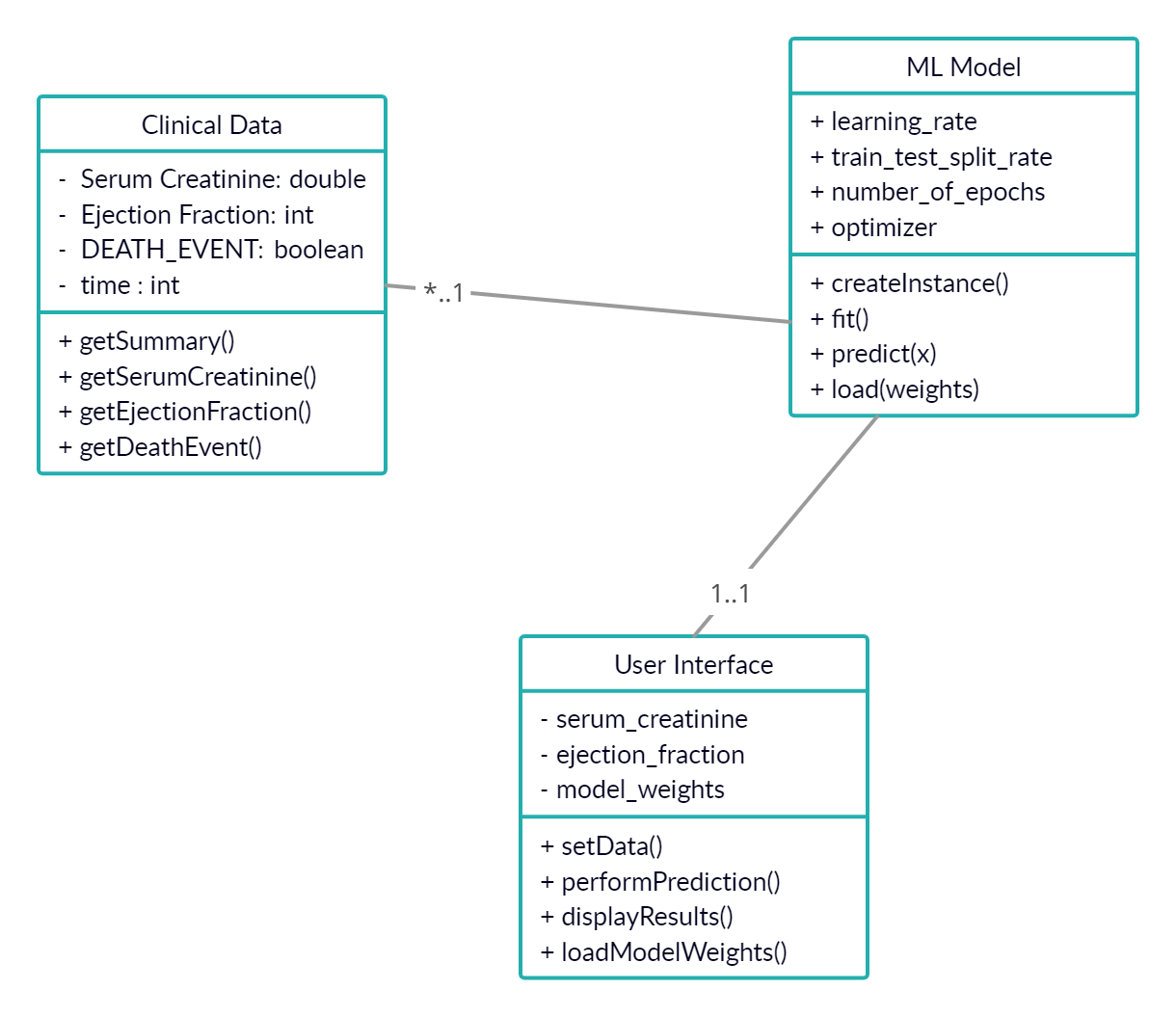
#### Class Diagram

Class diagrams are the main building blocks of every object-oriented method. The class diagram can be used to show the classes, relationships, interface, association, and collaboration. UML is standardized in class diagrams. Since classes are the building block of an application that is based on OOPs, so as the class diagram has appropriate structure to represent the classes, inheritance, relationships, and everything that OOPs have in its context. It describes various kinds of objects and the static relationship in between them.

The main purpose to use class diagrams are:

1. This is the only UML which can appropriately depict various aspects of OOPs concept.
2. Proper design and analysis of application can be faster and efficient.
3. It is base for deployment and component diagram.

Each class is represented by a rectangle having a subdivision of three compartments name, attributes and operation.



#### Fig 4.2.2.1 Class diagram

#### Sequence diagram

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modelling a new system.

The aim of a sequence diagram is to define event sequences, which would have a desired outcome. The focus is more on the order in which messages occur than on the message per se. However, the majority of sequence diagrams will communicate what messages are sent and the order in which they tend to occur.

#### Basic Sequence diagram notations Class roles and participants

Class roles describe the way an object will behave in context. Use the UML object symbol to illustrate class roles, but don't list object attributes.

#### Activation or Execution Occurrence

Activation boxes represent the time an object needs to complete a task. When an object is busy executing a process or waiting for a reply message, use a thin grey rectangle placed vertically on its lifeline.

#### Messages

Messages are arrows that represent communication between objects. Use half- arrowed lines to represent asynchronous messages.

Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks.

#### Lifelines

Lifelines are vertical dashed lines that indicate the object's presence over time.

#### Destroying Objects

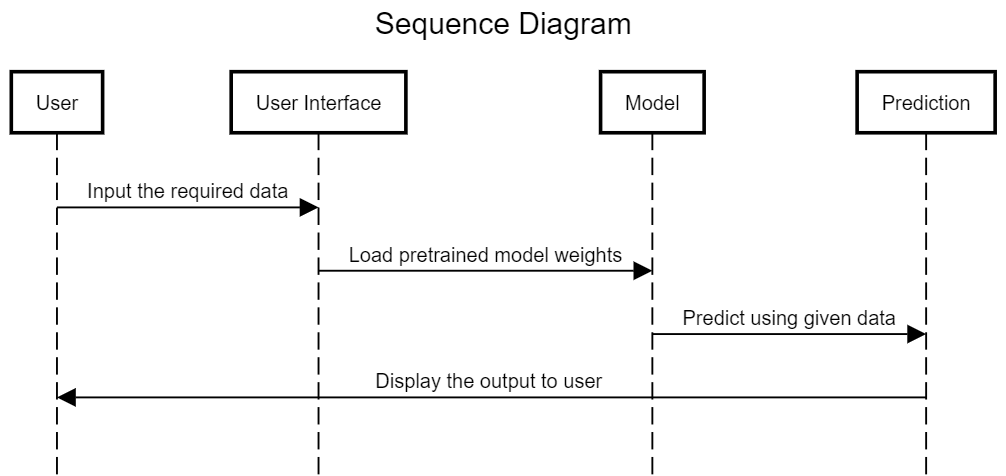
Objects can be terminated early using an arrow labelled "<< destroy >>" that points to an X. This object is removed from memory. When that object's lifeline ends, you can place an X at the end of its lifeline to denote a destruction occurrence.

#### Loops

A repetition or loop within a sequence diagram is depicted as a rectangle. Place the

condition for exiting the loop at the bottom left corner in square brackets []. Guards

When modelling object interactions, there will be times when a condition must met for a message to be sent to an object. Guards are conditions that need to be used throughout UML diagrams to control flow.



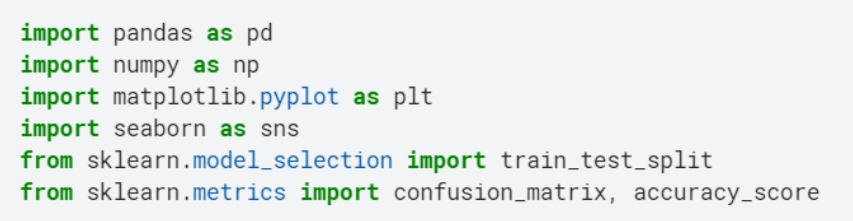
**Fig 4.2.3.1 Sequence diagram**

# CHAPTER - 5

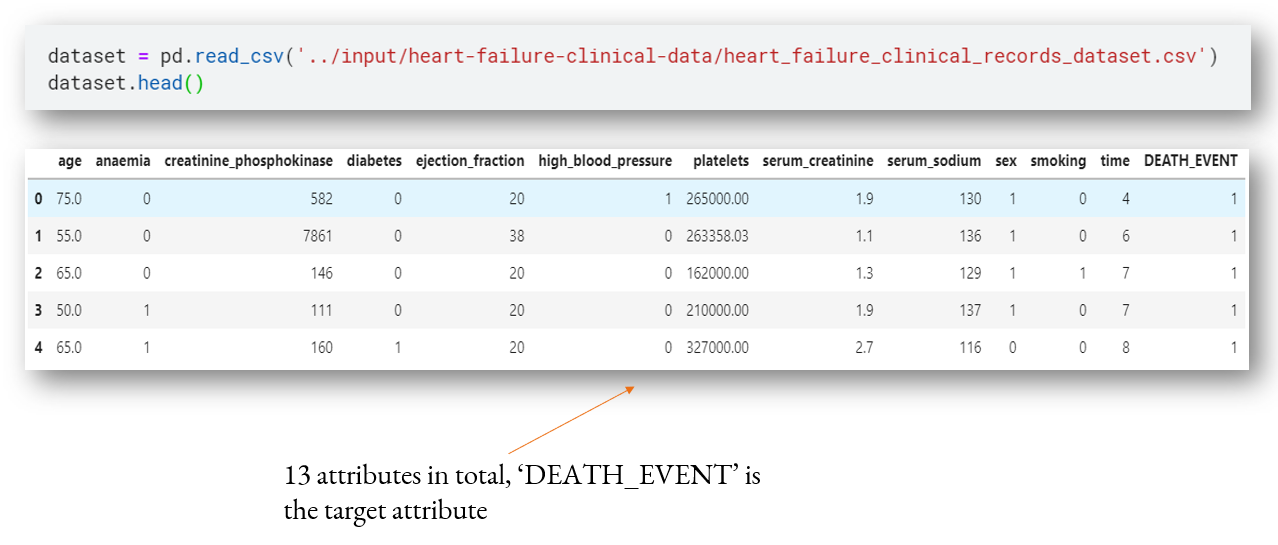
## IMPLEMENTATION

### Code Snippets

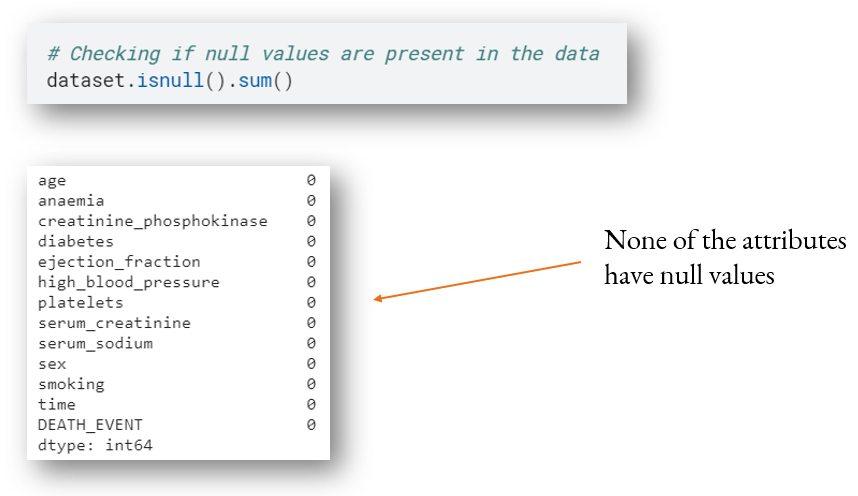
#### 5.1.1 Importing required libraries



#### 5.1.2 Loading the dataset



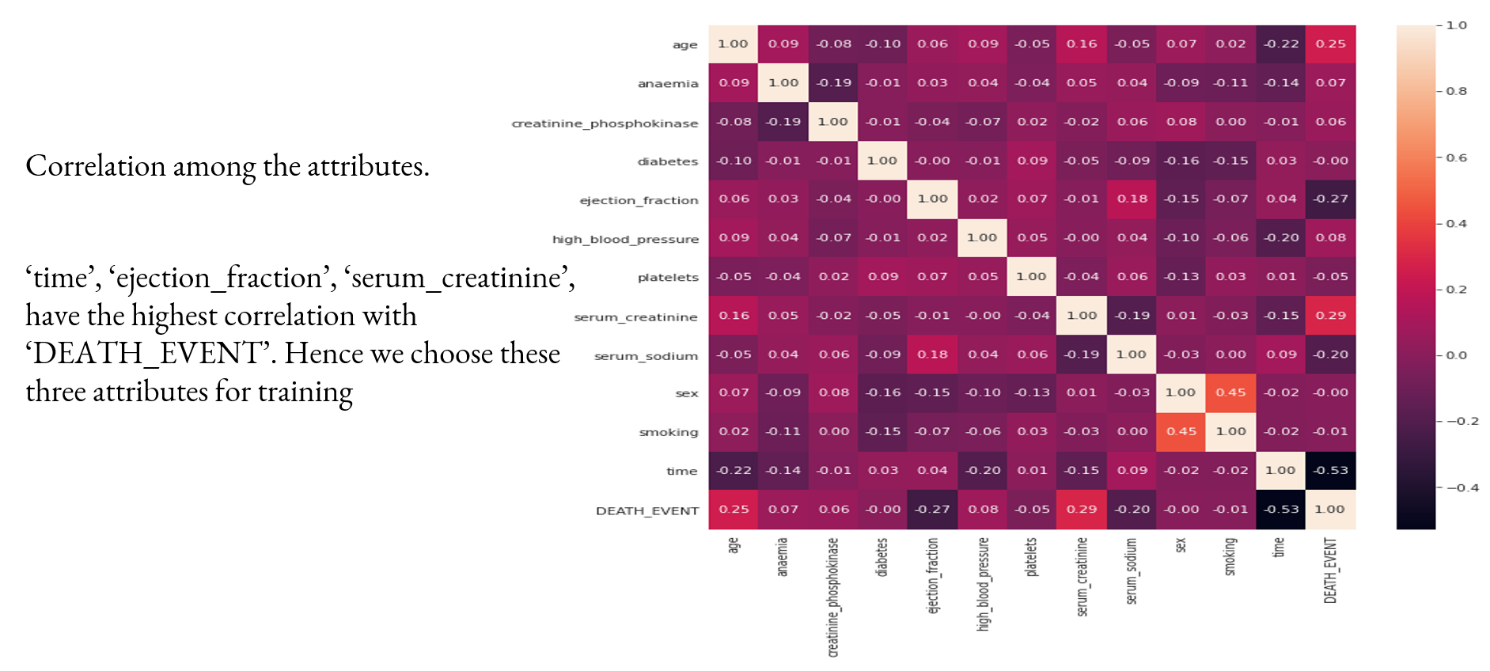
#### 5.1.3 Preprocessing



#### 5.1.4 Feature Selection



# 



#### 5.1.5 Data Visualization

#### 5.1.5.1 Analysis on Survival – Anaemia

# 

#### 5.1.5.2 Analysis on Survival – High Blood Pressure

# CHAPTER - 6

* 1. **Introduction to Testing**

## TESTING

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

According to ANSI/IEEE 1059 standard, Testing can be defined as - A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.

#### Who does Testing?

It depends on the process and the associated stakeholders of the project(s). In the IT industry, large companies have a team with responsibilities to evaluate the developed software in context of the given requirements. Moreover, developers also conduct testing which is called Unit Testing. In most cases, the following professionals are involved in testing a system within their respective capacities:

* + Software Tester
  + Software Developer
  + Project Lead/Manager
  + End User

Levels of testing include different methodologies that can be used while conducting software testing. The main levels of software testing are:

* + Functional Testing
  + Non-functional Testing

#### Functional Testing

This is a type of black-box testing that is based on the specifications of the software that is to be tested. The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for. Functional testing of a software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

#### Software Testing Life Cycle

The process of testing a software in a well-planned and systematic way is known as software testing lifecycle (STLC).

Different organizations have different phases in STLC however generic Software Test Life Cycle (STLC) for waterfall development model consists of the following phases.

1. Requirements Analysis
2. Test Planning
3. Test Analysis
4. Test Design

#### Requirements Analysis:

In this phase testers analyze the customer requirements and work with developers during the design phase to see which requirements are testable and how they are going to test those requirements.

It is very important to start testing activities from the requirements phase itself because the cost of fixing defect is very less if it is found in requirements phase rather than in future phases.

#### Test Planning

In this phase all the planning about testing is done like what needs to be tested, how the testing will be done, test strategy to be followed, what will be the test environment, what test methodologies will be followed, hardware and software availability, resources, risks etc. A high level test plan document is created which includes all the planning inputs mentioned above and circulated to the stakeholders.

#### Test Analysis

After test planning phase is over test analysis phase starts, in this phase we need to dig deeper into project and figure out what testing needs to be carried out in each SDLC phase. Automation activities are also decided in this phase, ifautomation needs to be done for software product, how will the automation be done, how much time will it take to automate and which features need to be automated. Nonfunctional testing

areas(Stress and performance testing) are also analyzed and defined in this phase.

#### Test Design

In this phase various black-box and white-box test design techniques are used to design the test cases for testing, testers start writing test cases by following those design techniques, if automation testing needs to be done then automation scripts also needs to written in this phase.

### Test Cases

Test Cases are divided into 3 parts based on 3 modules Student, Admin, Cell Members.

1. Student Test Cases
2. Admin Test Cases
3. Cell Member Test Cases

#### Student Test Cases

* + - * The student must be able to register before lodging any complaint.
      * The student must be able to login with their credentials.
      * The student must be able to lodge the complaint and upload file (if any).
      * The student must be able to track the status of the complaints.

#### Admin Test Cases

* + - * The admin must be able to see all the complaints.
      * The admin must be able to add / manage cell members.
      * The admin must be able to add / manage students.
      * The admin must be able to update the status of the complaints.
      * The admin must be able to see the user logs.

#### Cell Member Test Cases

* + - * The cell member must get complaints of their respective category only.
      * The cell member must be able to update the status of the complaint.
      * The cell member must be able to forward the complaint if wrong category is selected.

### Test case 1: User registration

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test scenario ID** | | Registration-1 | | | **Test case ID** | | | Registration-1A | | |
| **Test case description** | | User registration | | | **Test case priority** | | | High | | |
| **Pre-requisite** | | Username, password | | | **Post-requisite** | | | User dashboard | | |
| **Test Execution steps** | | | | | | | | | | |
| **S.N**  **o** | **Action** | | **Inputs** | **Expected output** | | **Actual Output** | **Test Browser** | | **Test Result** | **Test comments** |
| 1 | Launch Application | | https://localhost/vsc ms/register | Register page | | Login page | Google Chrome | | Pass | Application successfully launched |
| 2 | Enter Username, Email & Password and hit  signup button | | Username: user123 Email id:  [user123@gmail.co](mailto:user123@gmail.com) [m](mailto:user123@gmail.com)  Password: User123$ | Redirect to login page | | Redirect to login page | Google Chrome | | Pass | Registration successful |

**Test case 2: User login**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test scenario ID** | | Login-1 | | | **Test case ID** | | | Login-1A | | |
| **Test case description** | | User login – Positive case | | | **Test case priority** | | | High | | |
| **Pre-requisite** | | User email id | | | **Post-requisite** | | | Username, password | | |
| **Test Execution steps** | | | | | | | | | | |
| **S.N**  **o** | **Action** | | **Inputs** | **Expected output** | | **Actual Output** | **Test Browser** | | **Test Result** | **Test comments** |
| 1 | Launch Application | | https://localhost/vsc ms | Login page | | Login page | Google Chrome | | Pass | Application successfully launched |
| 2 | Enter username, password and hit login button | | Username: user123 Password: Abc123$ | Redirect to dashboard | | Redirect to dashboard | Google Chrome | | Pass | Login successful |

### Test case 3: User login

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test scenario ID** | | Login-1 | | | **Test case ID** | | | Login-1B | | |
| **Test case description** | | User login – Negative case | | | **Test case priority** | | | High | | |
| **Pre-requisite** | | User email id | | | **Post-requisite** | | | Username, password | | |
| **Test Execution steps** | | | | | | | | | | |
| **S.N**  **o** | **Action** | | **Inputs** | **Expected output** | | **Actual Output** | **Test Browser** | | **Test Result** | **Test comments** |
| 1 | Launch Application | | https://localhost/vsc ms | Login page | | Login page | Google Chrome | | Pass | Successfully logged in |
| 2 | Enter username, password and hit login button | | Username: user123 Password: Student123$ | An alert with message “The  details you entered is/are  incorrect” | | An alert with message  “The details you entered is/are  incorrect” | Google Chrome | | Pass | Invalid login attempt stopped |

**Test case 4: Register complaint**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test scenario ID** | | Lodge-1 | | | **Test case ID** | | | Lodge-1A | | |
| **Test case description** | | Register complaint – Positive case | | | **Test case priority** | | | High | | |
| **Pre-requisite** | | Logged in student dashboard | | | **Post-requisite** | | | Registered a complaint | | |
| **Test Execution steps** | | | | | | | | | | |
| **S.N**  **o** | **Action** | | **Inputs** | **Expected output** | | **Actual Output** | **Test Browser** | | **Test Result** | **Test comments** |
| 1 | Login as student | | Clicks on register complaint | Register complaint page | | Register complaint page | Google Chrome | | Pass | Register complaint page works |
| 2 | Enter required details and hit register  button | | Complaint details | Complaint successfully registered | | Complaint successfully registered | Google Chrome | | Pass | Complaint successfully registered |

### Test case 5: View Complaints

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test scenario ID** | | Complaints-1 | | | **Test case ID** | | | Complaint-1A | | |
| **Test case description** | | View all complaints registered by the student | | | **Test case priority** | | | High | | |
| **Pre-requisite** | | Logged in student dashboard | | | **Post-requisite** | | | Complaints fetched | | |
| **Test Execution steps** | | | | | | | | | | |
| **S.N**  **o** | **Action** | | **Inputs** | **Expected output** | | **Actual Output** | **Test Browser** | | **Test Result** | **Test comments** |
| 1 | Enter user credentials and hit Login button | | Username, password | Student dashboard | | Student dashboard | Google Chrome | | Pass | Logged in successfully |
| 2 | Click on My complaints | | Clicks on My complaints | My complaints page | | My complaints page | Google Chrome | | Pass | My complaints page works |

**Test case 6: Update complaint status**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test scenario ID** | | Status-1 | | | **Test case ID** | | | Status-1A | | |
| **Test case description** | | Update complaint status | | | **Test case priority** | | | High | | |
| **Pre-requisite** | | Logged in admin dashboard with at least 1 complaint registered | | | **Post-requisite** | | | Complaint status updated | | |
| **Test Execution steps** | | | | | | | | | | |
| **S.N**  **o** | **Action** | | **Inputs** | **Expected output** | | **Actual Output** | **Test Browser** | | **Test Result** | **Test comments** |
| 1 | Click on view complaint details | | NA | Complaint details page | | Complaint details page | Google Chrome | | Pass | Complaint details page works |
| 2 | Click on Take action | | NA | Update complaint page | | Update complaint page | Google Chrome | | Pass | Update complaint page works |
| 3 | Selects a complaint status, enters a response and hits update  button | | Complaint status, remark | Complaint status updated successfully | | Complaint status updated successfully | Google Chrome | | Pass | Complaint status updated successfully |

**Test case 7: Forward a complaint**

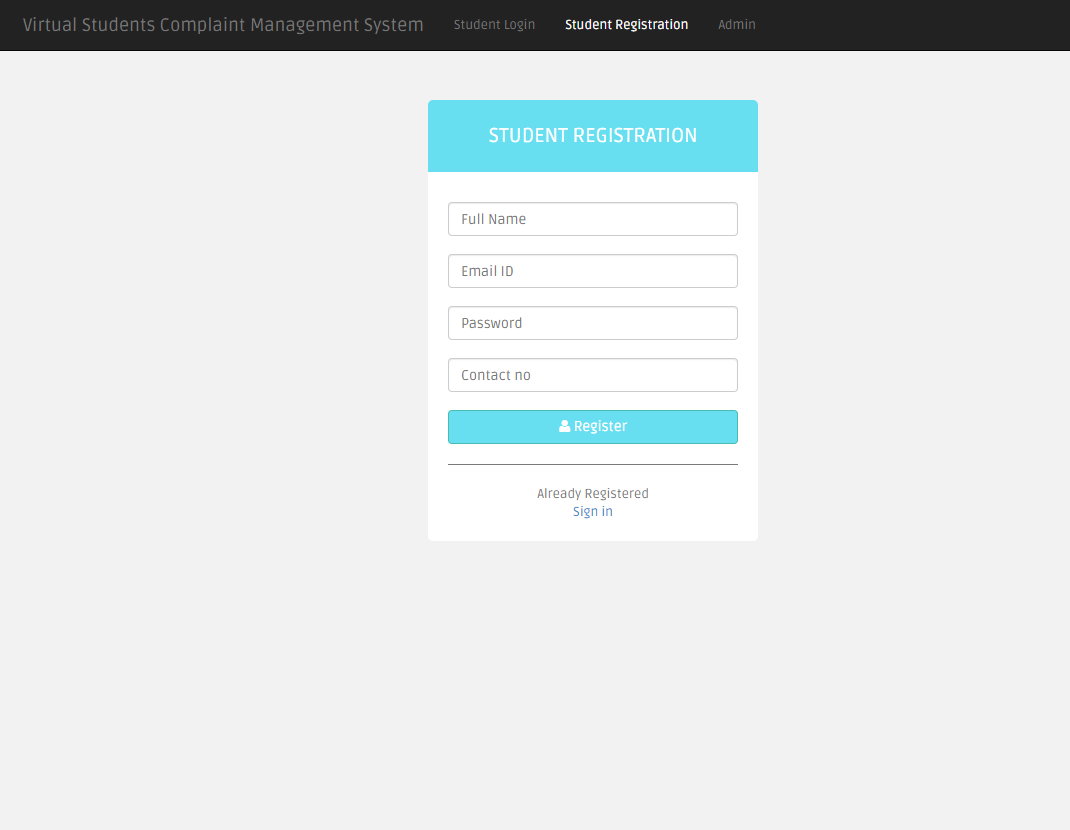
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test scenario ID** | | Forward-1 | | | **Test case ID** | | | Forward-1A | | |
| **Test case description** | | Forward a complaint | | | **Test case priority** | | | High | | |
| **Pre-requisite** | | Logged in admin/cell member dashboard with at least 1  complaint registered | | | **Post-requisite** | | | Complaint forwarded | | |
| **Test Execution steps** | | | | | | | | | | |
| **S.N**  **o** | **Action** | | **Inputs** | **Expected output** | | **Actual Output** | **Test Browser** | | **Test Result** | **Test comments** |
| 1 | Click on view complaint details | | NA | Complaint details page | | Complaint details page | Google Chrome | | Pass | Complaint details page works |
| 2 | Click on Forward To | | NA | Update complaint page | | Update complaint page | Google Chrome | | Pass | Update complaint page works |
| 3 | Selects a category (to be fprwarded to)  and hits forward button | | Category to be forwarded to | Complaint forwarded successfully | | Complaint forwarded successfully | Google Chrome | | Pass | Complaint forwarded successfully |

# CHAPTER - 7

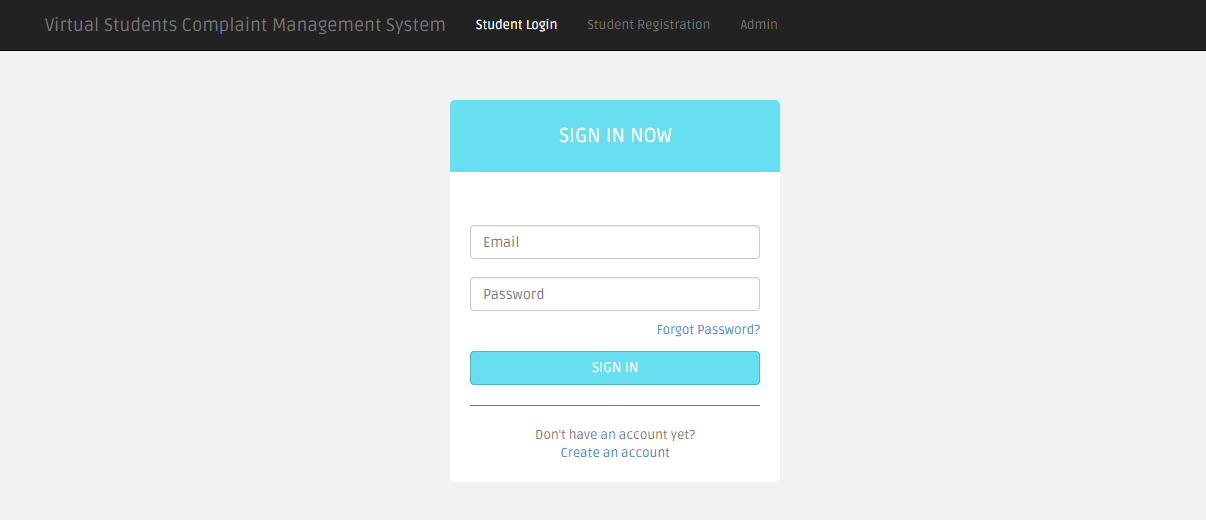
## SCREENSHOTS

****

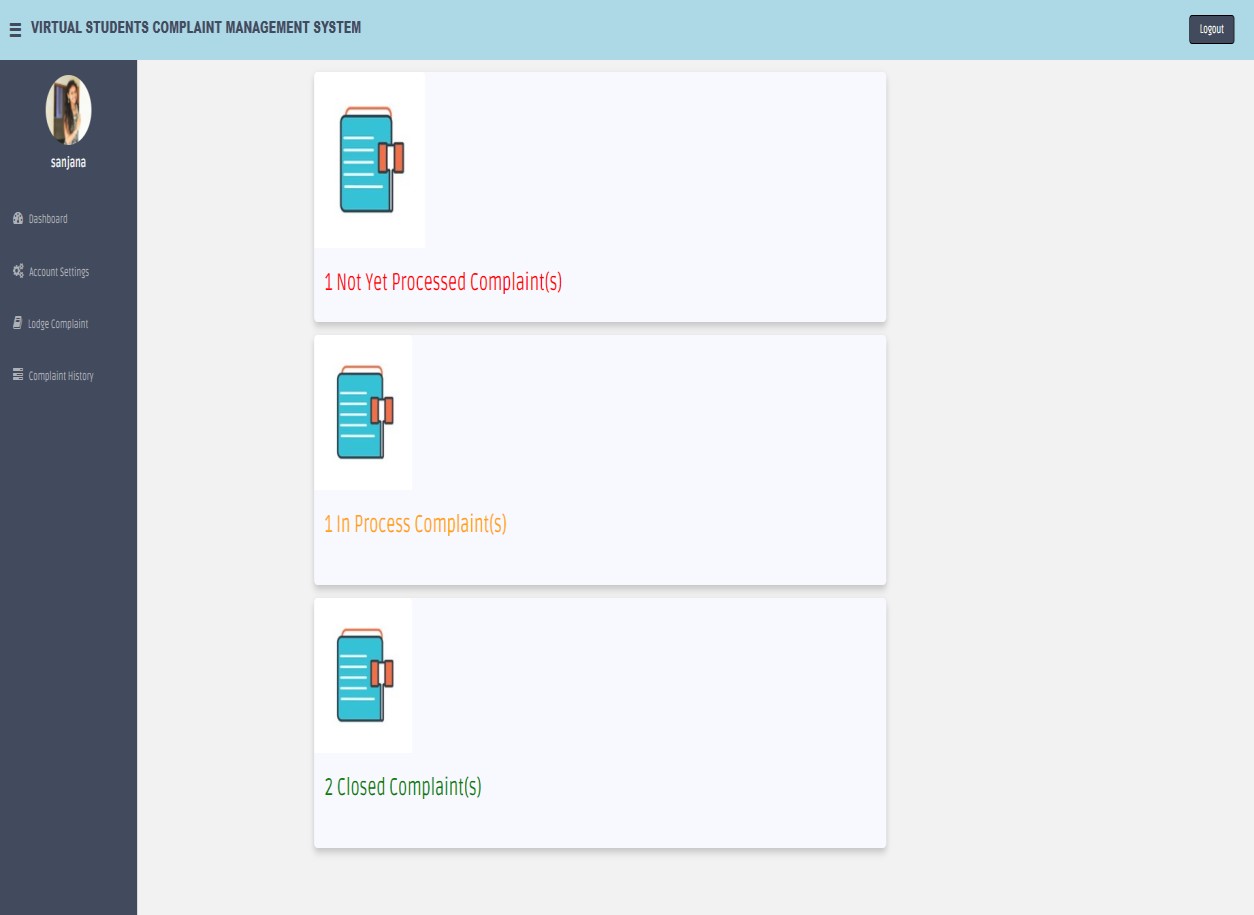
#### Fig 7.1 Home Page

****

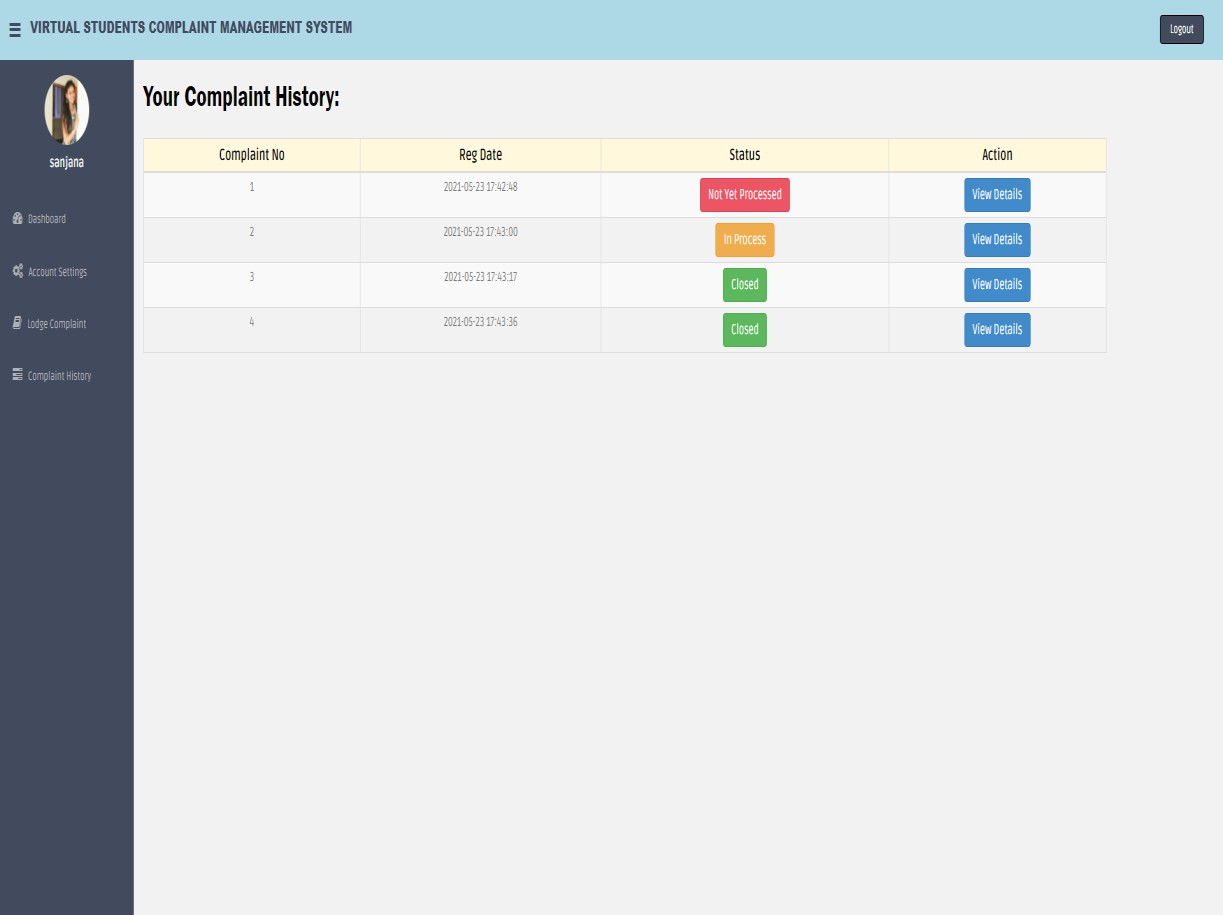
**Fig 7.2 Student Registration**



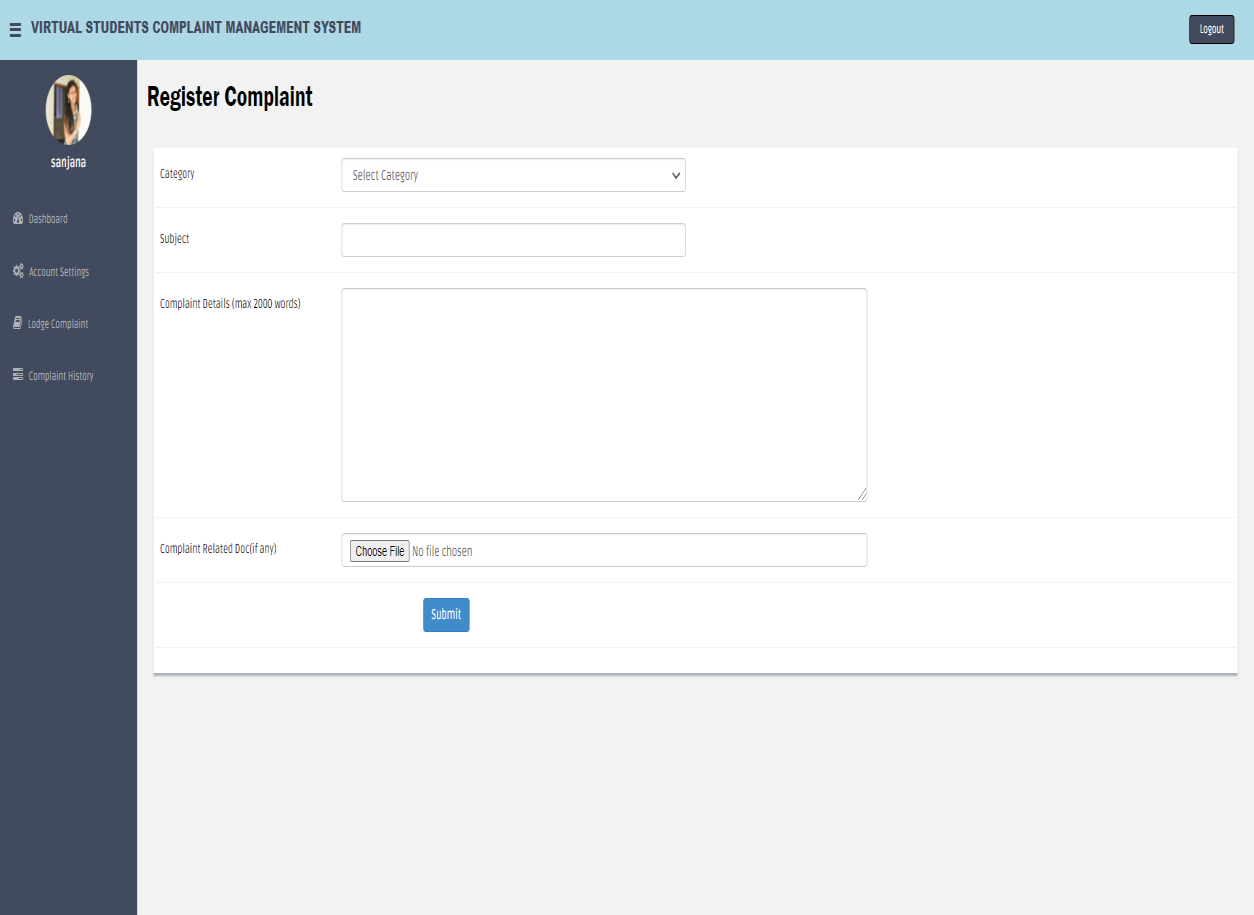
#### Fig 7.3 Student Login

****

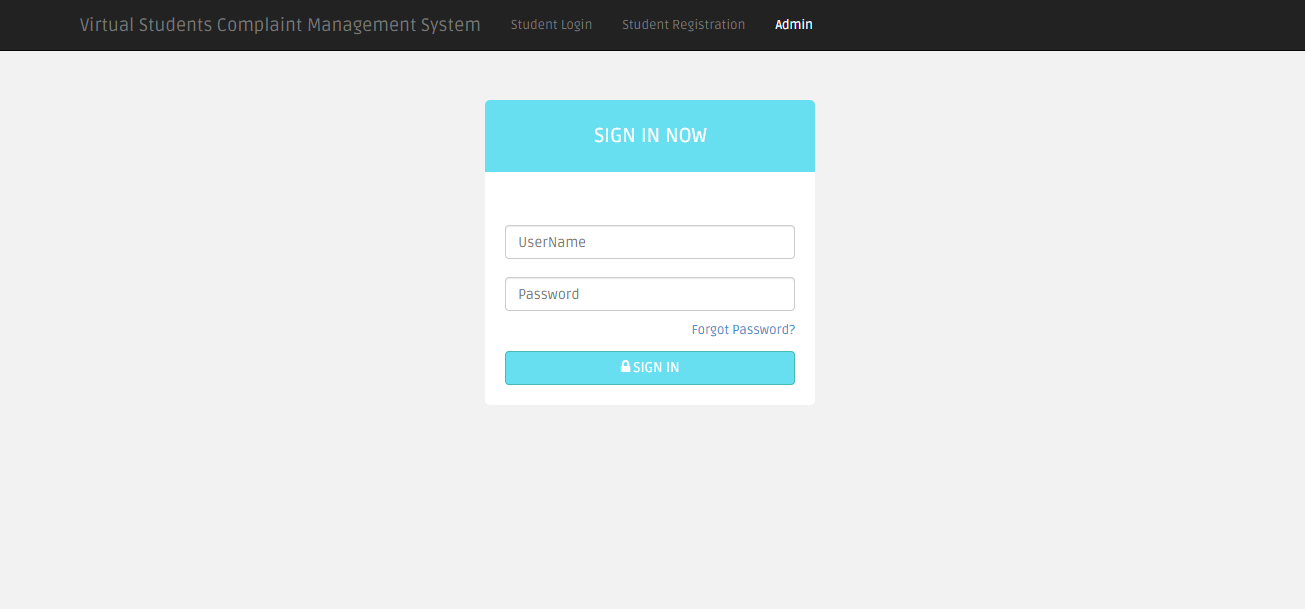
**Fig 7.4 Student Dashboard**



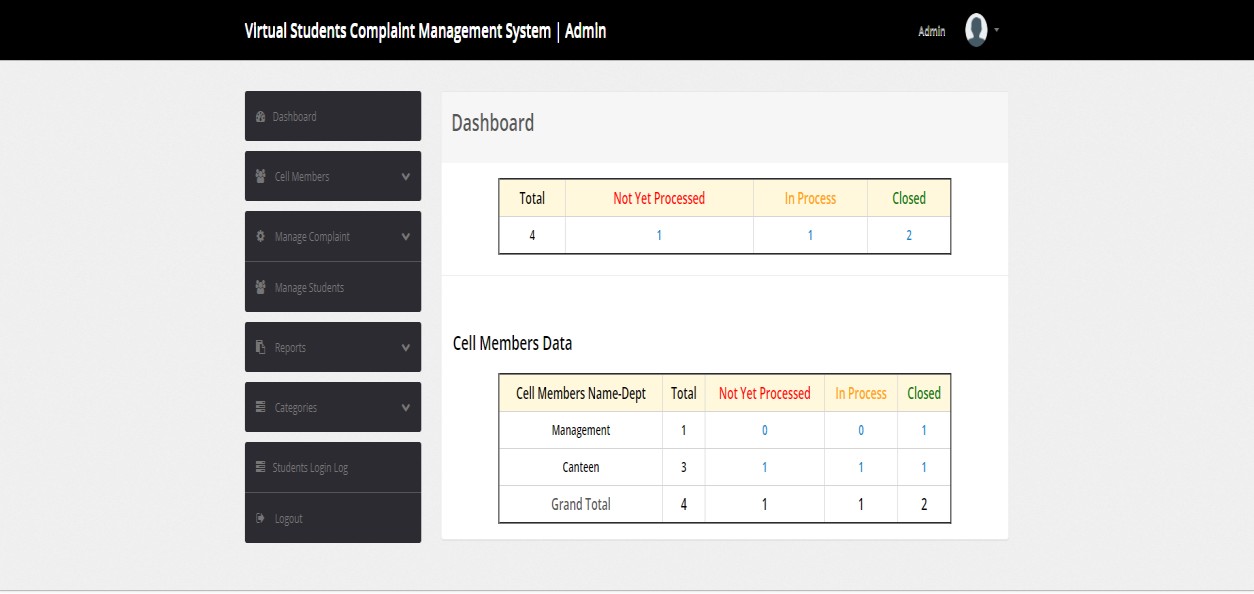
#### Fig 7.5 Complaint History

****

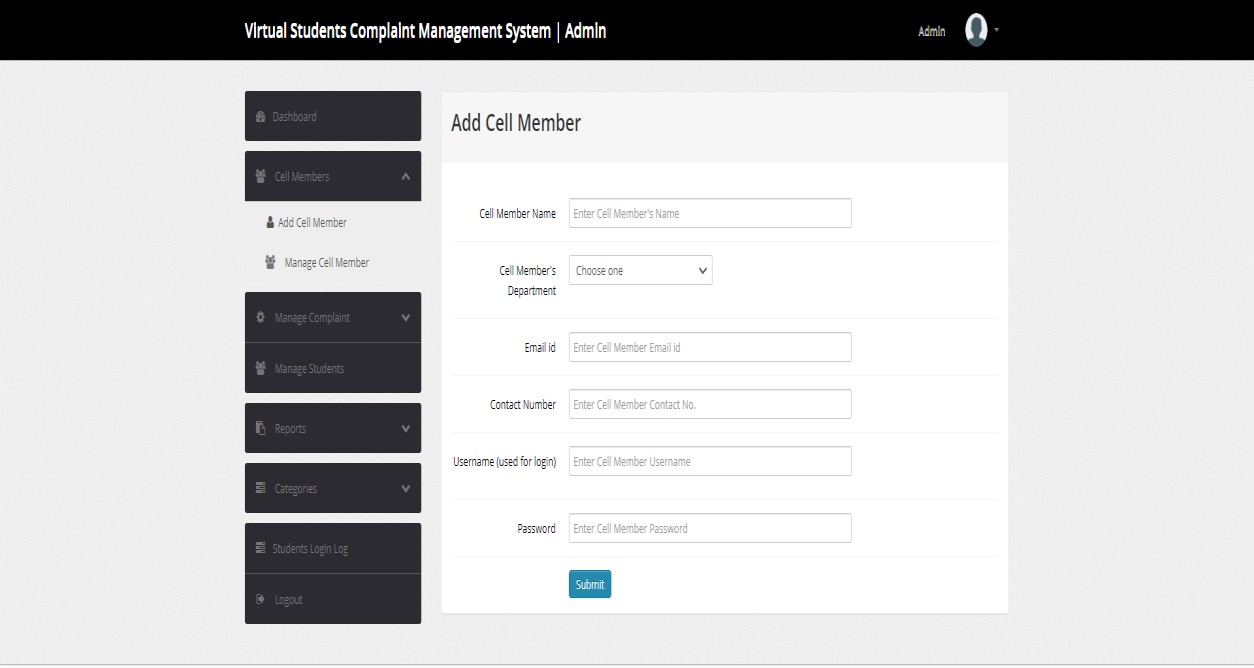
**Fig 7.6 Register Complaint**



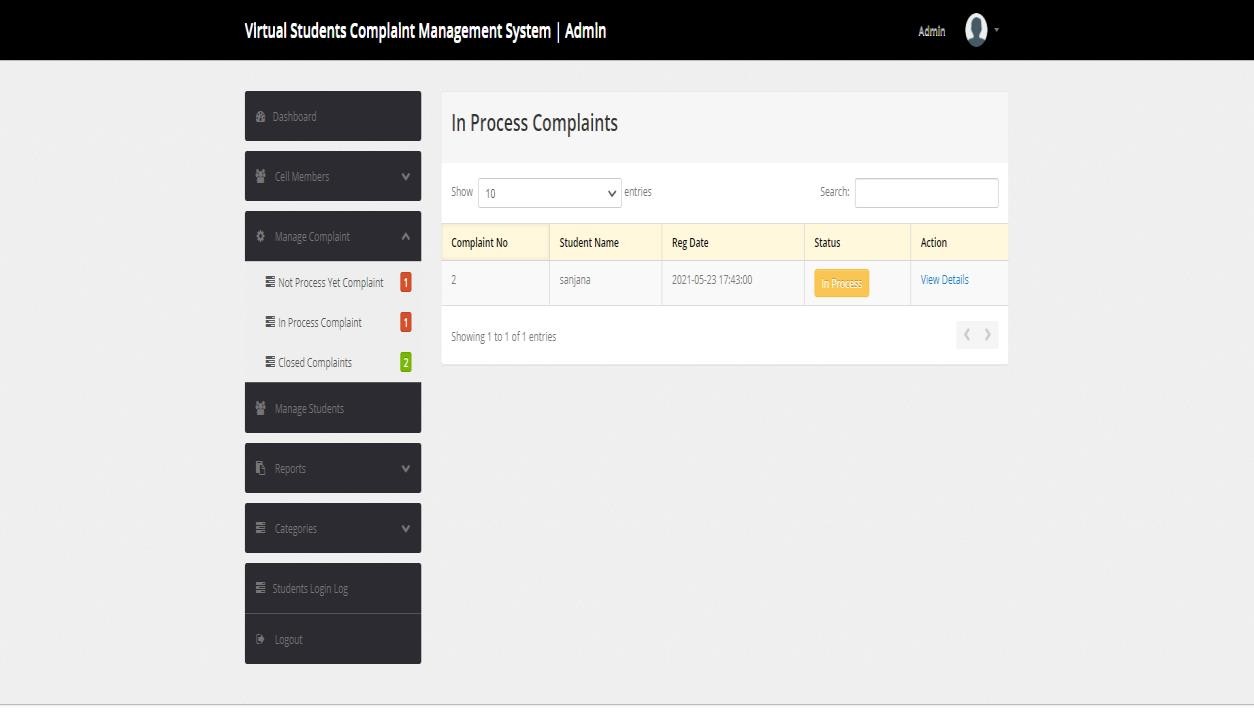
#### Fig 7.7 Admin Login

****

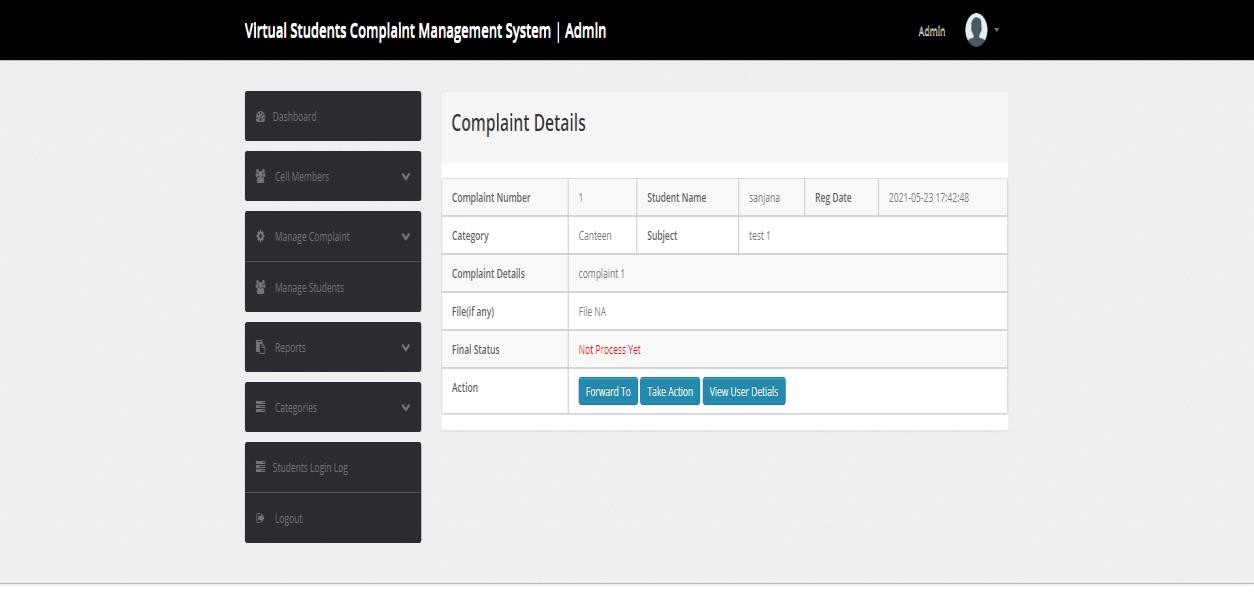
**Fig 7.8 Admin Dashboard**



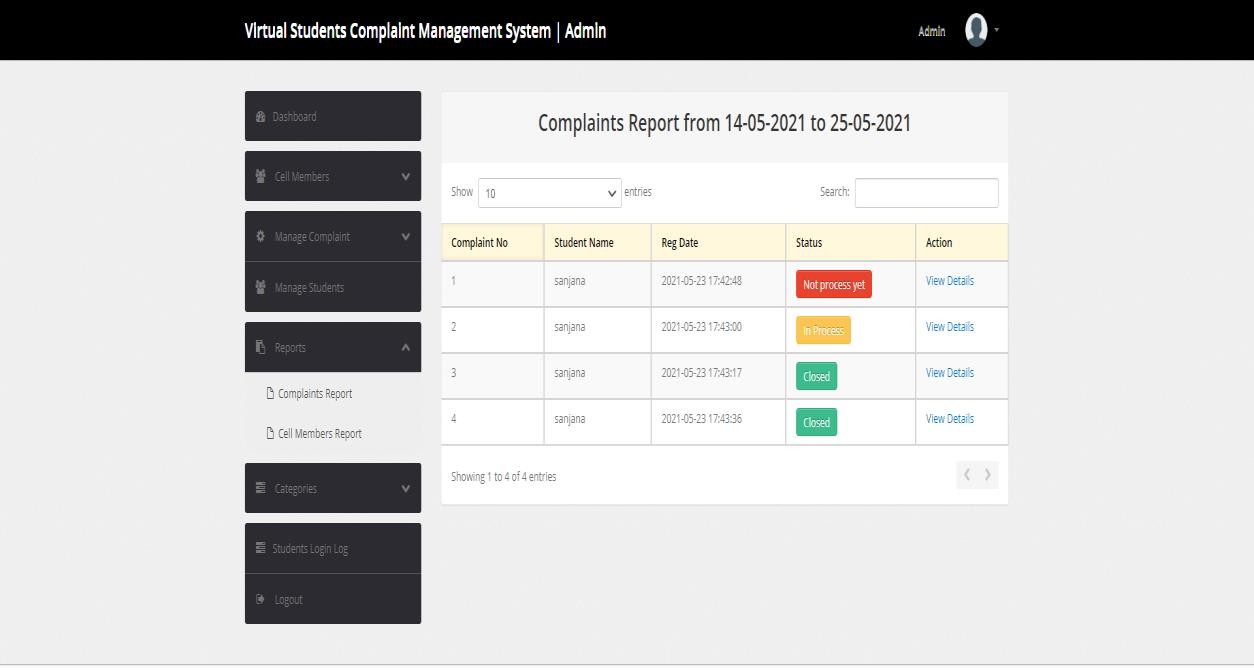
#### Fig 7.9 Add Cell Member

****

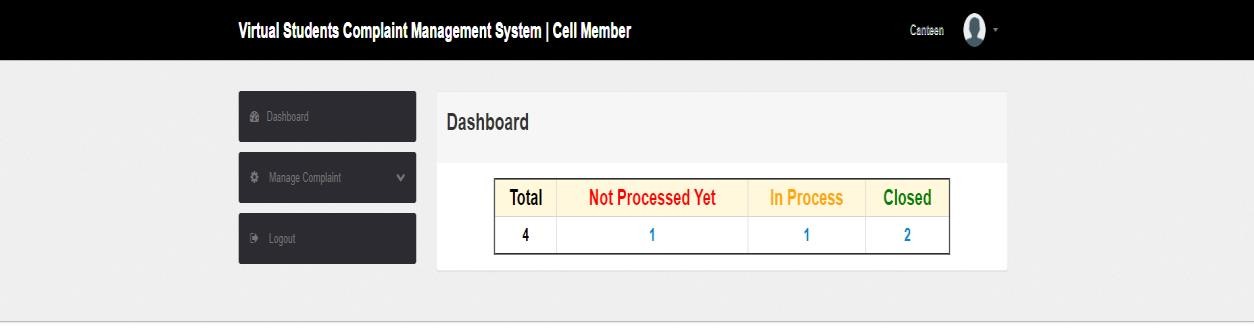
**Fig 7.10 All In Process complaints**



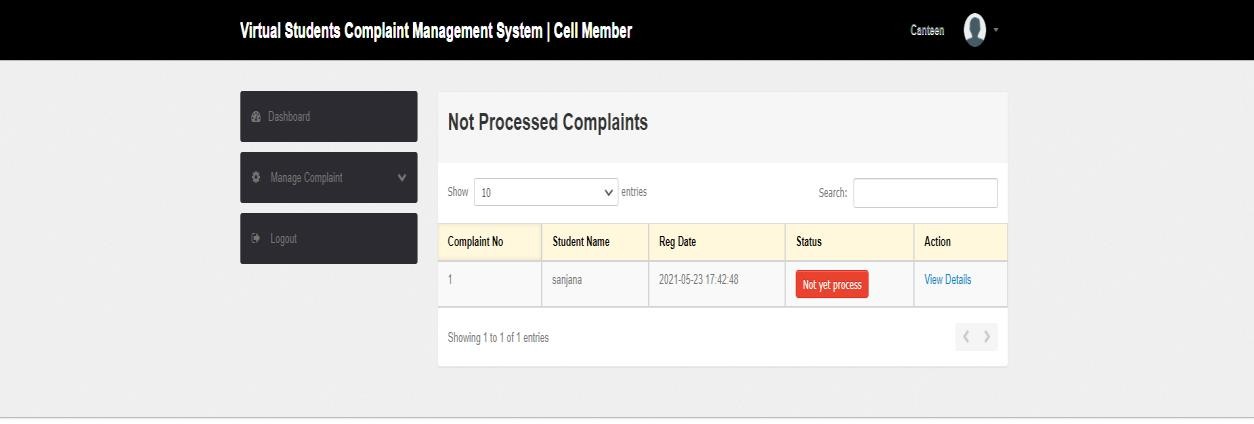
#### Fig 7.11 Complaint Details

****

**Fig 7.12 Complaints Report**



#### Fig 7.13 Cell Members Dashboard

****

**Fig 7.14 Not Processed Complaints**

# CHAPTER - 8

## FUTURE ENHANCEMENTS

Progressive web applications are very flexible and can be integrated with various technologies. Therefore, many significant improvements can be made to the project. Following are the features that can be added to the project:

#### Receive notifications:

Current system works on the principle of checking complaints on a daily/weekly basis. Instead, notifications can be sent to Admin/Cell member whenever a complaint is registered, or a complaint is forwarded in their respective department. Also, notifications can be sent to student when his/her complaint status is updated.

#### ML based algorithms to route complaints to their categories:

Currently, a student has to select a category for his complaint while registering the same. Instead, an algorithm can be used to sort complaints based on keywords and route them to respective departments.

#### Interaction between Admin and student:

Currently, an admin/cell member can post a response while updating complaint status. What if admin/cell members needs some additional information about the complaint, or if there is a query on provided details ? Admin/Cell member can send a message to the students asking for required information and student should be able to post a reply for the same.

# CHAPTER - 9

## CONCLUSION

A Grievance or Complaint can be defined as any kind of communication that expresses discontent about an action or lack of action, about the standard of service or absence of service of an institute and the complainant asks for remedial action. Virtual Students Complaint Management System is necessary in all educational organizations to investigate reason for the dissatisfaction of student and to obtain a speedy resolution to the problem. It also helps to take appropriate action ensuring the promises that are kept by the institution.

This Complaint Management System is used to establish a fair, independent and consistent system for redressal of various issues faced by the students in private or public institutions and to develop a responsive attitude among all, thereby maintaining harmonious atmosphere in the campus, with students facing a minimal delay.

# CHAPTER - 10

## REFERENCES

1. Shaligram Prajapat, Vaibhav Sabharwal, Varun Wadhwani, “A Prototype for Grievance Redressal System”, 2017
2. Mukesh Buldak, Shrikant Pandhekar, Afzal Gigani, Amreshsinh Kachwah, Kundan Patil, Poonam Polshetwar, Pradeep Jadhav, “Journal Paper on Online Grievance Redressal System”, 2019
3. Osman A. Nasr, Enwa Ali, “Online Complaint Management Systems”, June 2015
4. International Journal of Science and Research (IJSR) 2(6):305-307
5. Pattamaporn Kormph, Panida Liawsomboon, Narut Phongoen, Siripen Pongpaiche, University of California, Irvine, Smart Complaint Management System, July 2018, Conference: 2018 Seventh ICT International Student Project Conference (ICT-ISPC)
6. Kroenke, David, MIS Essentials (Fourth ed.), Pearson, Boston, p. 10, 2015.
7. Bulgacs, Simon, "The first phase of creating a standardised international technological implementation framework/software application". Int. J. Business and Systems Research. Vol.7, No.3, pp.250 – 265, 2013.
8. Gutte Avinash, Nikhil Kate, Ajayraj Hulikere, Shrikant Kokate, “E-College: An ERP for Educational Institute”, International Journal of Computer Sciences and Engineering, Vol. 2, Issue.3, pp.134-138, 2014.
9. Natalya Prokofyeva, Victoria Boltunova, “Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems”, December 2017, Procedia Computer Science 104:51-56.
10. Surag Shahu Gaikwadi , Prof Pratibha, PES Modern College of Engineering Pune, India, “A Review Paper on Bootstrap Framework”, 2019.
11. Ilam Baruthi, “Creating a Local Web Server Using XAMPP”, 2018.