1.**INTRODUCTION**

* 1. **Abstract**

Medicare is a web application in which customers can order their medicine by uploading the picture the of prescription and get the home delivery of medicine by cash on delivery. Medicare solves the problem for people who live in remote areas. The people will get the service available at any time the customer can order their medicines in 2 ways: By uploading the picture of the prescription and by searching for medicine. Our aim is to provide medicine to patients at the doorstep and aware customers of online medicine availability.

Medicare contains a medical dictionary in which the customer can search for medicine and output is generated on the basis of medicine name, content, rate, manufacturing details, and also company name.

Medicare enables online consultation. It can be achieved through chatting and video conferences. It helps patients to consult the doctor in the pandemic situation we are going through.

* 1. **Purpose**

The Medicare app solves the problem for people who live in remote areas. The people will get the services available at any time.

In the traditional appointment system patients has to come to the hospital and queue at the appointment window to make the appointment and difficult to purchase medicines from the pharmacy.

The current scenario is that the customers can upload the picture of the prescription and get the medicine by cash on delivery or by online payment.

And the patient can easily make their appointments through the application and it provides online consultation.

* 1. **Scope**

**MEDICARE** is a system through which a user or simply a patient can purchase the medicine online and it provides the online consultation. Our aim is to aware the customer for online medicine availability and the case of online appointments and consultations.

* 1. **Overview**

There are several online pharmacy and consultation tools in the marketplace. Some of which are not available in small cities or remote areas. Hence the customers are not aware of online medicine availability and online appointment scheduling and consultation.

There are five modules in this system. The first three modules are web application for the admin, doctor, and pharmacy. In which the customer can order their medicine by two ways: By uploading the picture of prescription and by searching a medicine. Other four modules are websites for Admin, Hospital, Pharmacy, and Doctor. The admin can only grants the permission to the Hospital, Pharmacy, and Doctors to ensure the safety of customers.

**2. SYSTEM ANALYSIS**

**2.1 Existing System**

Currently, there is a application for the online pharmacy and consultation,. But it is not available in small cities or remote areas. Hence the customers are not aware about online medicine availability and online consultation.

* 1. **Limitations of Existing System**
* It is not available in small cities or remote areas.
* This is the application which do not provides the online consultation like netmeds.
* Time consuming.
  1. **Proposed System**

The proposed system is a web-application that allows one to have easy access to all the relevant information about online pharmacy, doctor appointment and consultation. Patient, admin , doctor, pharmacy, hospital have login section in our web-application. Admin can access the information about patient, hospital, pharmacy, doctor and medicine .Admin can control the overall system. Hospital is responsible for the doctor management and the pharmacy is responsible for the medicine management. Patient can also see the near by pharmacy and hospital.

* 1. Advantage of Proposed System

**MEDICARE**

* Time efficient.
* User friendly.
* Greater efficiency.
* Easy access to current information.
  1. **Hardware Requirements**

**2.6 Software Requirements**

* Operating system: windows 10
* Front end: HTML, CSS, BOOTSTRAP, XML
* Back end: Python(FLASK), SQLYOG, WAMP, Android Java
* IDE: Pycharm ,Andriod studio

**2.7 Feasibility Study**

Feasibility is defined as the practical extent to which a project can performed successfully. The objective of feasibility study is to establish the reason for developing the software that is acceptable to the users, adaptable to change the comfortable to the established standards.

**Various types of feasibility that are commonly considered include:**

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

Economic feasibility determines whether the proposed system is capable for generating financial gains for an organization. It involves cost incurred on the software development team, estimated cost of hardware, and cost of performing feasibility study and so on. The proposed system is economically feasible since the cost incurred for the development of the system produces long term gains. The cost of hardware and software for the class of application of society management system is less. Hence the proposed system is economically feasible.

1. **Technical Feasibility**

Technical feasibility access the current resource (include hardware and software) and technology which are required to accomplish the user requirements in the system within the allocated time and budget. It is concerned with the existing computer system (hardware and software) and to what extend it can support the proposed system.

1. **Operational Feasibility**

The proposed object would be beneficial to organization satisfying the objectives when developed and installed. One of the main problems faced during development of a new system is getting the acceptance for the user. There is support from the management of organization towards the development of the project. All the operational aspects are considered carefully. Thus the project is operationally feasible.

**3. SYSTEM DESIGN**

**3.1 USERS OF THE SYSTEM**

Admin

Hospital

Pharmacy

Doctor

Patient

**MODULES**

**Functions of Admin**

* The admin can log into the system.
* The admin can view hospital & clinic approved.
* The admin can view approved hospital & clinic.
* The admin can view pharmacy & approved.
* The admin can view approved pharmacy.
* The admin can give reply to the complaints reported by the users.
* The admin can view the doctors.
* The admin can view the numbers of users logged into the system.
* The admin can view the details of the patients such as name, email, image

**Functions of Hospital**

* **The hospital must sin up before logging into the application**
* **The hospital can view& update their profile**
* **The hospital can manage doctors**
* **The hospital can view doctors**
* **The hospital can view & update doctors schedule**
* **The hospital can view doctors booking**
* **The hospital can view patient booking**
* **The hospital can view & update doctors profile**
* **The hospital can add & view schedule**
* **The hospital can view patients**

**Functions of Pharmacy**

* The pharmacy must sign up before logging into the application
* The pharmacy can add medicine.
* The pharmacy can add & view stock.

• The pharmacy can view patient.

• The pharmacy can view sales report.

• The pharmacy can view booking & its sub info.

• The pharmacy can update their status.

• The pharmacy can add & view the discounts.

• The pharmacy can view others discount

* The pharmacy can view & update their profile

**Functions of Doctor**

**• The doctor can logging in the application**

**• The doctor can view & update their profile.**

**• The doctor can view the schedule.**

**• The doctor can view patent booking.**

**• The doctor can view patients.**

**• The doctor can view & upload prescription.**

**• The doctor can video conference & chat with patients.**

**Functions of Patient**

• The patient must sign up before the logging application.

• The patient can view & update their profile.

• The patient can book the doctors.

• The patient can view doctors.

• The patient can view medicine.

• The patient can upload prescription.

• The patient can view nearest pharmacy.

• The patient can buy medicine.

• The patient can view medicine discounts.

• The patient can send complaint.

• The patient can video conference & chat with doctor

**3.2 Requirement Modeling**

**Data Flow Diagram**

Data flow diagram is used to define the flow of the system and resources and such as information .Data low diagrams are a way of expressing system requirement in a graphical manner. DFD represents one of the most ingenious tools used for structure analysis. DFD is also known as bubble chart. It has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design.

In the normal, logical DFD can be completed using only four notations.

**Data Flow Diagram Symbols**

**External Entry**

**Dataflow**

**Process**

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Data Store**

* To construct a data flow diagram , we use
* Squares representing external entities, which are sources and destinations of data.
* Rounded rectangles representing processes, which take data as input, do something to it,

and output it.

* Arrows representing the data flows which can either, be electronic data or physical items.
* Open-ended rectangle representing data store including electronic store such as database or XML files and physical store such as or filling cabinets or stack of paper.

**PROCESS:**

A process transforms incoming data flow into outgoing data flow.

**DATA STORE:**

Data store are repositories of data in the system. They bare sometimes also referred to as file.

**DATA FLOW:**

Dataflow are pipelines through which packets of information flow. Label the arrow with the

name of he data that moves through it.

**EXTERNAL ENTITY:**

External entities are objects outside the system with which system communicates. External

Entities are sources destinations of the system inputs and outputs.

**DATA FLOW DIAGRAM LAYERS:**

Data flow diagram in several nested layers. A single process node on high level diagram can

be expanded to show a more detailed data flow diagram. Draw the content diagram first,

followed by various layers of data flow diagrams.

**LEVEL 0**



**LEVEL 1**

**LEVEL 1.1**



**LEVEL 1.2**



**LEVEL 1.3**



**LEVEL 1.4**



**LEVEL 1.5**



**3.3 NORMALIZATION**

Normalization is the process of organizing the fields and tables of relational database to minimize redundancy. Normalization usually involves dividing large tables into smaller tables and defining relationships between them. The objective is to isolate data so that additions, deletions and notification of a field can made in just one table and then propagated through the set of database using the defined relationship.

**3.3.1 First Normal Form**

The aim of first normal form (1NF) is to make the structure of relationship simple by making sure that it does not have data aggregates or repeating groups. It specifies that the domains of attribute are inclusive of only atomic values and that any attribute value in tuple must be individual value from the domain of that attribute.

**3.3.2 Second Normal Form**

Fully functional dependency forms the basis of the second normal form. Each non-key attribute is fully dependent functionality on the total candidate key and a relation is in the second normal form.(2NF) if and only if it is in 1NF.

**3.3.3 Third Normal Form**

Transitive dependency are removed from the second normal form

**3.4 Table**

**Login**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| login\_id | Int | primary key |
| user name | varchar() | User name |
| Password | varchar() | Password |
| type | varchar() | Admin/hospital/doctor/pharmacy |

**Doctor**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| doc\_id | Varchar | Primary key |
| doc\_name | Varchar | Doctor name |
| doc\_img | Varchar | Image |
| doc\_quali | Varchar | Qualification |
| doc\_exp | Varchar | Experience |
| doc\_deprt | Varchar | Department |
| ph\_number | Int | Phone number |
| doc\_email | Varchar | Doctor email |
| hos\_id | Int | Foreign key |
| login\_id | Int | Foreign key |
| Place | Varchar | Place |
| Post | Varchar | Post office |
| Pin | Int | Pin code |
| District | Varchar | District |
| Address | Varchar | Address of doctor |
| Fees | int | Primary key |

**Hospital**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| hos\_id | Int | Primary key |
| hos\_name | Varchar | Hospital name |
| ph\_number | Int | Phone number |
| Place | Varchar | Place |
| District | Varchar | District |
| Post | Varchar | Post office |
| Pin | Int | Pin code |
| e\_mail | Varchar | E mail |
| hos\_lat | Varchar | Latitude |
| hos\_long | Varchar | Longitude |
| login\_id | Int | Foreign key |

**Medicine**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| med\_id | Int | Primary key |
| med\_name | Varchar | Medicine name |
| Price | Varchar | Medicine price |
| Description | Vaechar | Description |
| med\_brand | varchar | Medicine brand |

**Pharmacy**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| phar\_id | Int | Primary key |
| phar\_name | Varchar | Pharmacy name |
| phar\_number | Int | Phone number |
| Place | Varchar | Place |
| Post | Varchar | Post office |
| District | Varchar | District |
| Pin | Int | Pin code |
| phar\_email | Varchar | Pharmacy email |
| phar\_lat | Varchar | Latitude |
| phar\_long | Varchar | Longitude |
| login\_id | int | Foreign key |

**Chat**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| chat\_id | Int | Primary key |
| from\_lid | Int | Foreign key |
| to\_lid | Int | Foreign key |
| Msg | Varchar | Message |
| Date | Date | Date |

**Complaint**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| com\_id | Int | Primary key |
| pat\_id | Int | Foreign key |
| Complaint | Varchar | Complaint |
| Replay | Varchar | Complaint replay |
| Date | Date | Complaint date |
| Status | varchar | Complaint status |

**Discount**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| dis\_id | Int | Primary key |
| med\_id | Int | Foreign key |
| dis\_ftime | Varchar | Discount from-time |
| dis\_ttime | Varchar | Discount to-time |
| pat\_id | Int | Foreign key |

**Doctor Booking**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| db\_id | Int | Primary key |
| sch\_id | Int | Foreign key |
| book\_date | Date | Booking date |
| book\_time | Varchar | Booking time |
| pat\_id | Int | Foreign key |
| Status | Varchar | Booking status |
| doc\_id | Int | Foreign key |

**Medicine Booking**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| medc\_id | int | Primary key |
| med\_id | int | Foreign key |
| Date | Date | Date |
| Price | varchar | Medicine price |
| pat\_id | int | Foreign key |

**Medicine Booking sub**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| medbook\_id | Int | Primary key |
| med\_id | Int | Foreign key |
| Date | Date | Booking date |
| Price | varchar | Medicine price |
| Time | varchar | Booking time |

**Prescription**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| pres\_id | Int | Primary key |
| doc\_id | Int | Freign key |
| pat\_id | Int | Freign key |
| Date | Date | Date |
| db\_id | Int | Foreign key |

**Schedule**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| sch\_id | Varchar | P |
| sch\_ftime | Varchar | schedule from-time |
| sch\_ttime | Varchar | schedule to -time |
| doc\_id | Int | foreign key |
| sch\_date | Date | schedule date |

**stock**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| stock\_id | varchar() | primary key |
| Quantity | Varchar | quantity |
| med\_id | Varchar | Foreign |
| pat\_id | Varchar | Foreign |

**Cart**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Descipion** |
| cart\_id | int | primary key |
| patt\_lid | int | foeign key |
| med\_lid | int | forien key |

**Payment**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Daa ype** | **Descipion** |
| paymen\_id | Int | Pimay key |
| med\_ook\_id | int | foeign key |
| date | date | date |
| time | varchar | time |
| amount | int | amount |
| type | varchar | doctor/medicine |

**Bank**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Daa ype** | **Descipion** |
| bank\_id | int | primary key |
| bank\_name | varchar | foeign key |
| acc\_no | varchar | account number |
| bank\_pin | varchar | pin number |
| bank\_balance | varchar | bank balance |

**3.5 INTERFACE DESIGN**

**3.5.1 Input Design**

Input design is the part of the overall system design that requires very careful attention and is the most expensive phase. It is the point of contact for the users with the computer system and so it itself is prone to error, if data going into the system is in correct then processing and output will magnify these errors . Objective during input design are as follows:

* cost effective method input
* Achieve high level accuracy
* Ensure that input is free of ambiguity
* Data recording
* Produce Data transfer in input form
* Data verification
* Transmitting data to computer
* Data correction

**3.5.2 Output Design**

An inevitable activity in the system design is the proper design of input and output in a form acceptable to the user .Outputs from the system is required primarily to communicate the result for later consultation

**The various types of outputs required by most systems are:**

**External outputs**: whose destination is outside the organization and which require special attention.

**Internal outputs** : whose destination is within the organization and which require careful design because they are user’s main interface within the computer

**Operational outputs**: whose use is purely within the computer.

**Interactive outputs**: which involve the user in communicating with the computer.

**3.6 TECHNOLOGY OVERVIEW**

**➢ PHP**

PHP is a general-purpose server-side scripting language originally designed for Web

development to produce dynamic Web pages. It is one of the first developed server-side

scripting languages to be embedded into an HTML source document, rather than calling an external file to process data. Ultimately, the code is interpreted by a Web server with a PHP processor module which generates the resulting Web page. It also has evolved to include a command-line interface capability and can be used in standalone graphical applications. PHP can be deployed on most Web servers and also as a standalone shell on almost every operating system and platform free of charge. A competitor Microsoft's Active Server Pages (ASP) server-side script engine and similar languages, PHP is installed on more than 20 million Web sites and 1 million Web servers.

PHP was originally created by” Rasmus Lerdorf” in 1995. The main implementation of PHP is now produced by The PHP Group and serves as the formal reference to the PHP language. PHP is free software released under the PHP License, which is incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.

While PHP originally stood for "Personal Home Page", it is now said to stand for "PHP:

Hypertext Pre-processor", a recursive acronym

**➢ MySQL**

MySQL pronounced either "My S-Q-L" or "My Sequel," is an open source relational

database management system. It is based on the structure query language (SQL), which is used for adding, removing, and modifying information in the database. Standard SQL

commands, such as ADD, DROP, INSERT, and UPDATE can be used with MySQL.

MySQL can be used for a variety of applications, but is most commonly found on Web

servers. A website that uses MySQL may include Web pages that access information from a database. These pages are often referred to as "dynamic," meaning the content of each page is generated from a database as the page loads. Websites that use dynamic Web pages are often referred to as database-driven websites.

Many database-driven websites that use MySQL also use a Web scripting language like PHP to access information from the database. MySQL commands can be incorporated into the PHP code, allowing part or all of a Web page to be generated from database information. Because both MySQL and PHP are both open source (meaning they

free to download and use), the PHP/MySQL combination has become a popular choice for database-driven websites.

**➢ HTML**

Hypertext Markup Language (HTML) is the main markup language for web pages. HTML

elements are the basic building-blocks of Webpages.

HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like<html>), within the web page content. HTML tags most commonly come in pairs like <h1>and</h1>, although some tags, known as empty elements, are unpaired, for example <img >.The first tag in a pair is the start tag, the second tag is the end tag(they are also called opening tags and closing tags). In between these tags web designers can add text, tags, comments and other types of text-based content.

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts in languages such as JavaScript which affect the behavior of HTML webpages. S

Web browsers can also refer to Cascading Style Sheets (CSS) to define the appearance and layout of text and other material. The W3C, maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicitly presentational HTML markup.

**Elements**

HTML documents are composed entirely of HTML elements that, in their most general form have three components: a pair of tags, a "start tag" and "end tag"; some attributes within the start tag; and finally, any textual and graphical content between the start and end tags, perhaps including other nested elements. The HTML element is everything between and including the start and end tags. Each tag is enclosed in angle brackets.

**Attributes**

Most of the attributes of an element are name-value pairs, separated by "=" and written within the start tag of an element after the element's name. The value may be enclosed in single or double quotes, although values consisting of certain characters can be left unquoted in HTML (but not XHTML). Leaving attribute values unquoted is considered unsafe. In contrast with name-value pair attributes, there are some attributes that affect the element simply by their presence in the start tag of the element.

**➢ CSS**

Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation

semantics (the look and formatting) of a document written in a markup language. It’s most common application is to style web pages written in HTML and XHTML, but the language can also be applied to any kind of XML document, including plain XML, SVG and XUL.

CSS is designed primarily to enable the separation of document content (written in HTML or a similar markup language) from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design). CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links thatdocument to a CSS style sheet, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified

CSS specifies a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities or weights are calculated and assigned to rules, so that the results are predictable.

**➢PYTHON**

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Python comes with a large standard library that supports many common programming tasks such as connecting to web servers, searching text with regular expressions, reading and modifying files. Python’s interactive mode makes it easy to test short snippets of code. There's also a bundled development environment called IDLE.Is easily extended by adding new modules implemented in a compiled language such as C or C++.Can also beembedded into an application to provide a programmable interface. Python runs anywhere, including Mac OS X, Windows, Linux, and UNIX, with unofficial builds also available for Android and iOS. Python is free software in two senses. It doesn't cost anything to download or use Python, or to include it in your application. Python can also be freely modified and re-distributed because while the language is copyrighted it's available under open source license.

**FLASK**

Flask is a web application framework written in Python. It is developed by Armin Ronacher, who leads an international group of Python enthusiasts named Pocco. Flask is based on the Werkzeug WSGI toolkit and Jinja2 template engine. Flask is a micro web framework written in Python. It is classified as a micro framework 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 is considered more Pythonic than the Django web framework because in common situations the equivalent Flask web application is more explicit. Flask is also easy to get started with as a beginner because there is little boilerplate code for getting a simple app up and running. Flask was originally designed and developed by Armin Ronacher as an April Fool's Day joke in 2010. Despite the origin as a joke, the Flask framework became wildly popular as an alternative to Django projects with their monolithic

**PyCharm IDE 2017.1.2**

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, Web, and data science development. It is developed by the Czech company Jet Brains .It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda. PyCharm is cross-platform, with Windows, macOS and Linux versions. PyCharm provides API so that developers can write their own plugins to extend PyCharm features. Several plugins from other Jet Brains IDE also work with PyCharm. There are more than 1000 plugins which are compatible with PyCharmstructure and dependencies.

Jet Brains has developed PyCharm as a cross-platform IDE for Python. In addition to supporting versions 2.x and 3.x of Python, PyCharm is also compatible with Windows, Linux, and macOS. At the same time, the tools and features provided by PyCharm help programmers to write a variety of software applications in Python quickly and efficiently. The developers can even customize the PyCharm UI according to their specific needs and preferences. Also, they can extend the IDE by choosing from over 50 plug-ins to meet complex project requirements.

PyCharm makes it easier for developers to implement both local and global changes quickly and efficiently. The developers can even take advantage of the refactoring options provided by the IDE while writing plain Python code and working with Python frameworks. They can avail the rename and move refactoring for files, classes, functions, methods, properties, parameters, and local/global variables. Likewise, they can improve code quality by extracting variables, fields, constants, and parameters. Also, PyCharm allows programmersto break up longer classes and methods through extract method. PyCharm makes it easier for programmers to write various web applications in Python supporting widely used web technologies like HTML, CSS, JavaScript, Typescript and CoffeeScript. The web developers can use the live editing preview option provided by the IDE to view a single web page simultaneously in the editor and browser. At the same time, the live edit feature provided by the IDE enables programmers to see the changes made to the code instantaneously on a web browser. PyCharm further allows developers to avail a JavaScript debugger as well as CoffeeScript and Type Script editors. It even simplifies isomorphic web application development by supporting both Angular JS and Node JS.

**WAMP**

Sands for “Windows, Apache, MySQL, and PHP”. XAMP is a variation o LAMP for Windows system and is often installed as a software bundle(Apache, MySQL, and PHP).It is often used for web development and internal ttesting, but may also be used to serve live websites.

The most important part of the XAMP packges is Apache (or “Apache HTTP server”) which is used run the we sever wihin Windows. By unning a local Apache web server on a Windows machine, a web developer can test Web pages in a Web brouser without publishing them live on the Inerrnet.

XAMP also includes MySQL and PHP, which are two o the most common technologies used for creating dynamic websites. MySQL is a high-speed database, while PHP is a scripting language that can be used to access data from the database. By installing these two components locally, a developer can build and test a dynamic website befforre publishing it to a public web server.

While Apache, MySQL, and PHP are open source components that can be installed individually, they are usually installed together. One popular package is called “XAMP Server”, which provides a user friendly way to install and configure the “AMP” components on Windows

**4. SYSTEM TESTING**

Testing the process of executing website is controlled manner in order to answer the suggestion dose the website behave be specified?” Testing is often used in association with the terms verification and validation. Verification is checking or testing of items, including websites, for conformance and consistency with associated specifications.

**Unit Testing**

Unit testing is a method by which individual unit of source code, set of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit to use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure.

**Module Level Testing**

In module level testing error will be found at each individual unit, it encourages the programmer to find and rectify the errors with affecting other modules.

**Integration Testing**

integration testing identifies problems that occur when unit is combined. By using a test plan that requires you to test each unit and ensure the viability of each before combining unit. You know that any error discovered when combining units likely related to the interface between units. This method reduce the number of possibilities to far simpler level of analysis

**Validation Testing**

It ensures that a product has been build according to requirements and design specifications. Validation ensures that “you built the right product”. In this majority of validations done during the data entry in the operation where there is a maximum possibility of entering wrong data.

**Output Testing**

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box

testing or system tests. Black Box testing methods focus on the functional requirement of the

software. That is, Black Box testing enables the software engineer to derive sets of input

conditions that will fully exercise all functional requirements for a program. Black Box

testing attempts to find errors in the following categories; incorrect or missing functions,

interface errors, errors in data structures or external database access, performance errors and

initialization errors and termination errors.

**5. SYSTEM IMPLEMENTATION**

**4.1 Introduction**

System implementation is the final stage of software development life cycle. For the successful implementation and cooperating of new system user must be selected, educated and trained. Unless the users are not trained, the system will be complex and it will feel as a burden of them. A software implementation method is a systematically structured approach to effectively integrate software based service or component into the workflow of an organizational structure or an individual end-user. A software implementation method is a blue print to get users and organization running with specific software product. The method is a set of rules and views to scope up with the most common issue that occur when implementing a software product: business alignment from the organizational view and the acceptance from the human view. It is stated as implementation of software consumes up to 1/3 of the budget of a software purchase. The complexity of implementing product differs on several issues. Examples are: the number of end users that will use the product, the effects that the implementation’s has on changes of task and responsibilities for the end users the culture and the integrity of the organization where the software is going to be used and thebudget available for acquiring the software.

**6. CONCLUSION**

**CONCLUSION**

The project report entitled” Medicare“ has come to its final stage. The system has been developed with much care that it is free of errors and at the same time it is efficient and less time consuming. The important thing is that the system is robust. We have tried our level best to make the site as dynamic as possible. Also provision is provided for future developments in the system. The entire system is secured. This online system will be approved and implemented soon.

**SCOPE FOR FUTURE ENHANCEMENT**

The project has a very vast scope in future this project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion.

As feature enhancement, we can add more detailed information about hospital, pharmacies, doctors and medicine. We can add the ambulance facility and other health care products, beauty tips, etc.

**7. BIBLIOGRAPHY**