Software Solution for Augment Curation and Patient Explorer

Major project report submitted in partial fulfilment of the requirement for the degree of **Bachelor of Technology**

in

Computer Science and Engineering

By

Kritika lodha 181398

Under the supervision of

Dr. Yugal Kumar

Department of Computer Science Engineering and Information Technology
Jaypee University of Information Technology, Waknaghat, 173234, Himachal Pradesh, INDIA

CERTIFICATE

Candidate's Declaration

I hereby declare that the work presented in this report entitled **"Software solution for Augment Curation and Patient Explorer "** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from Feb 2022 to May 2022 under the supervision of **Dr Yugal Kumar, Associate Professor** .
The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature) -

Name : Kritika Lodha
Rollno : 181398

This is to certify that the above statement made by the candidate is true to the best of my knowledge

(College Supervisor Signature)
Supervisor Name
Designation
Department name
Dated

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for his divine blessing that made it possible to complete the project work successfully.

I am quite grateful to my supervisor, Dr. Yugal Kumar, Department of CSE Jaypee University of Information Technology, Waknaghat, for her assistance. To complete this assignment, my supervisor has extensive knowledge and a deep interest in the subject of Web development. His never-ending patience, intellectual direction, constant encouragement, constant and energetic supervision, constructive criticism, good suggestions, and reading many poor versions and fixing them at all stagesmade it possible to finish this job.

I'd like to thank Dr.Yugal Kumar, Department of CSE, for her invaluable assistance in completing my project.

I would also like to express my gratitude to everyone who has directly or indirectly assisted me in making this project a success. In this unique scenario, I'd want to appreciate the different staff members, both teaching and non-teaching, who have developed their helpful assistance and facilitated my project. Finally, I must express my gratitude for my parents' unwavering support and patience.

Kritika Lodha

Table of Contents

LIST OF ABBREVATION

- ML - Machine Learning
- AI - Artificial Intelligence
- SDLC - Software Development Life Cycle
- AWS – Amazon Web Services
- QC - Quality Control
- BI - Business Intelligence
- API - Application Programming Interface
- FTP - File Transfer Protocol
- SOAP - Simple Object Access Protocol
- MQL - Mongo Query Language
- HDFS - Hadoop Distributed File System
- DB – Database
- REST - Representational State Transfer

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 About company

Nference is developing a comprehensive software platform to synthesise the world's rapidly expanding but segregated biological information. nference employs cutting-edge neural networks for real-time, automated insight extraction from the biomedical literature, as well as large scale molecular and real-world datasets, by triangulating unstructured and organised information. The platform offers a wide range of applications in the life sciences ecosystem, from R&D through commercial strategy and operations.

The headquarters of Nference are located in Kendall Square (Cambridge, MA), the world's biotech capital. Nference has offices in Rochester, Minnesota, Bangalore, India, and Toronto, Canada (Canada). With 150+ employees who primarily hold advanced degrees from prestigious academic institutions such as the Massachusetts Institute of Technology (MIT), Harvard College, and Harvard Medical School, Nference is regarded as having one of the most distinguished scientific and engineering teams in the Biopharma industry by senior executives. The team is split 50:50 between technologists (computer science and artificial intelligence engineers, software developers, mathematicians, and statisticians) and biomedical scientists (PhD in Biology/Omics or MD/PhD physician scientists). Over 150 peer-reviewed, high-impact articles from the nference research team have been published, including in Science, Nature,National Academy of Sciences (PNAS), Cell, Proceedings of the, Journal of Biological Chemistry, and the New England Journal of Medicine (NEJM).

According to the Washington Post, Nference is known as the Google of Biomedicine.

1.2 Domain

As part of this organization, I started up in the domain of Backend Development. Then I shifted to full stack development to cater the needs of the high priority existing project.

I directly report to Chandan Mahto (cmahto@nference.net) - Tech Lead.

1.3 Understanding why are data and data collection important

They need to comprehend the relevance of information and information collecting to understand statistical engineering. Information and knowledge, data and information Raw information indicates the facts and facts that are daily used by an organisation. All sales are reported in retail. But an agency learns little by looking at every sale on the stage of men or women. After processing data to provide context, purpose and relevance, it is converted to data. In view of the fact that these are its greatest components every Wednesday, a business may also realise the need for additional cakes on a Tuesday.

1.3.1 Importance of Data

It is important for enterprise enterprises to be able to test and verify and act on the statistics. The pace of the change requires companies to respond quickly to the ever-changing demands of clients and external factors. Although short time spent may be necessary, decisions are step by step baffling, as companies in a worldwide business environment contend. Managers may also have to review large quantities of information before they can select key choices. Powerful business insight (BI) gadgets assist dynamic managers.

1.3.1 Business Intelligence

BI tools offer humans with the important info from the statistics that they must make significant selections. Organization facts are frequently placed away in one-of-a-kind, disconnected programming packages and databases. BI apparatuses acquire and procedure facts from special assets. They deliver an account of the information to expand the facts on leaders. Successful BI assists companies with distinguishing improvement openings, recognise purchaser tendencies and increment seriousness. BI is excellent while it's miles added in smooth to apply designs, as an instance, scorecards and dashboards.

1.4 Understanding what is meant by Software Engineering

Software engineering is described as the process of evaluating user needs, then re-designing, creating, and software testing to meet the needs.

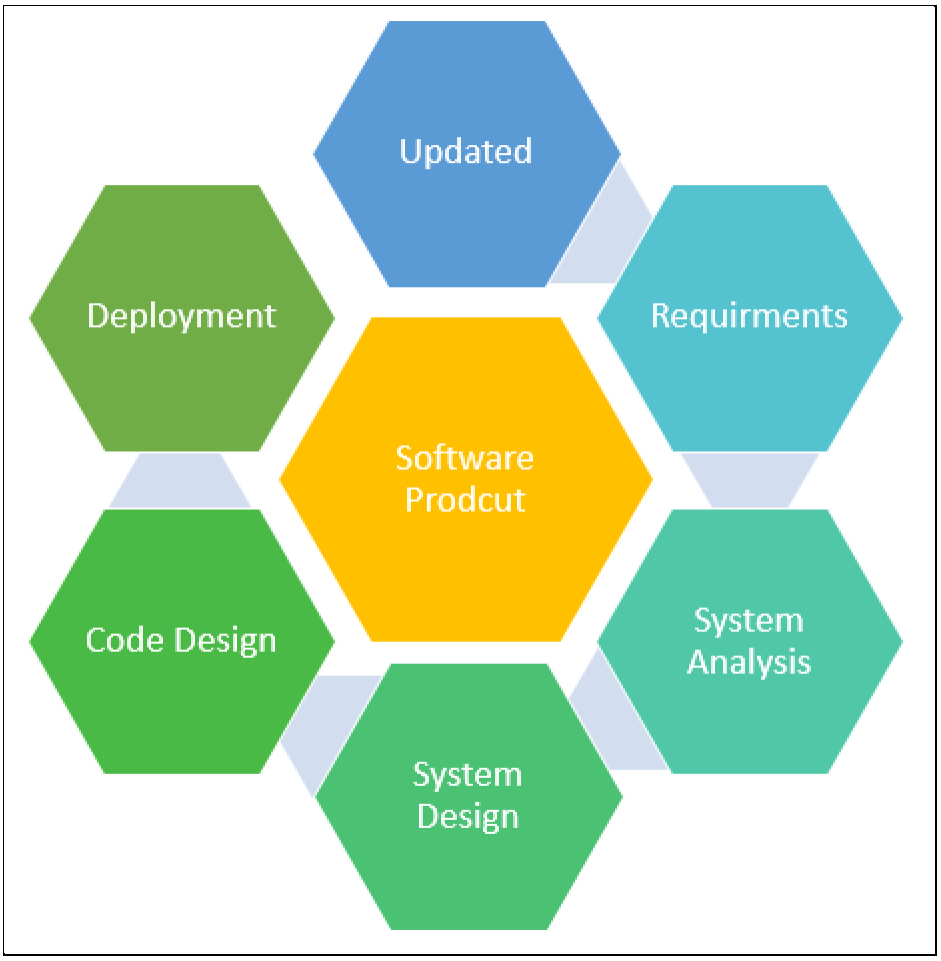

Fig 1. Constituents of a software product - adapted from [12].

Large software: In real life, it is far more convenient to construct a wall than it is to construct a house or structure. Similarly, when the size of the programme grows larger, software engineering aids in the development of the software.

Cost: The hardware sector has demonstrated its abilities, and large-scale production has reduced the price of computer and electrical gear.

Scalability: It is simpler to recreate new software to scale an older one if the process of software development is based on technical and scientific notions.

Adaptability: When the software development process is founded on scientific and engineering principles, it is simple to re-create new software using software engineering.

Quality Management: Provides a better approach of software development in order to provide high-quality software.

1.5 The role of a Software Engineer

In order to get high-quality and often updated information sets, it's crucial to differentiate between records pipelines that are achieved and wiped clean with the aid of software engineers. Software engineers are in charge of doing research,development, implementation, and maintenance of software systems.

Software engineers generally handle information units based and unstructured — in order to record the structure and applications, they should finally be educated with one of the types. The statistics engineer's toolkit also has a vast type of record technology, which includes the ever-growing variety of open source data input and processing frameworks.

To carry out their work in the programming languages such as Java, Scala C++, Python, Ruby, and databases like SQL and MongoDB software engineers may be expected to have these skills. They also require a strong level of expertise about extracting, reworking, loading equipment, and REST-oriented APIs to support the development and handling of integration activities for statistical purposes.

CHAPTER 2

LITERATURE SURVEY

In the field of software engineering, if they begin to discuss the topics about the tools and technologies which are used for the purpose of development and creating data pipelines, the number can go endless.

But, rather if they are to understand, the tools and technologies which are today used in the industry majorly, they can have a rational argument and understanding about the basic requirements which are at the core of most of the created products.

In Nference, the organization heavily relies on using open-sourced tools and technologies wherever it can be applied. Thus, following are some of the basic literature surveys of tools and technologies which are highly applicable and used by me.

2.1 Software Development

The development of software refers to a series of software-built, designed, implemented and supported computing processes.
A set of instructions or call it programmes instructs a computer to perform is called software itself. It is hardware-independent and programmable to computers.

Three fundamental kinds are available:

- System software for essential operations like operating systems, disc management, utilities, hardware and others.
- Software for programmers to construct tools like compilers and debuggers with text linkers and editors.
- Software application (apps or application) to support users in carrying out tasks. Examples include office data management software, media player, productivity suites, security tools. Apps could mean mobile and web applications such as Facebook, Google, Amazon.com etc.

2.1.1 Software Development Life Cycle

SDLC is a procedure / good practice used by the software technology sector in the design, development and testing of quality software. Software Development Life Cycle (SDLC) The SDLC aims at producing high quality software (apps) that satisfies or surpasses customers' expectations of a software, is delivered in time and cost estimates. SDLC describes tasks performed at every stage in the development of software. ISO/IEC 12207 is an international standard in order to assure the software life cycle process. The standard aims to define all the tasks necessary for the creation and maintenance of software. SDLC is a software project in a software organisation followed by a process. It gives a clear strategy on how certain software may be created, maintained, replaced and updated. The life cycle outlines a strategy to improve the quality of the software and its development processes.

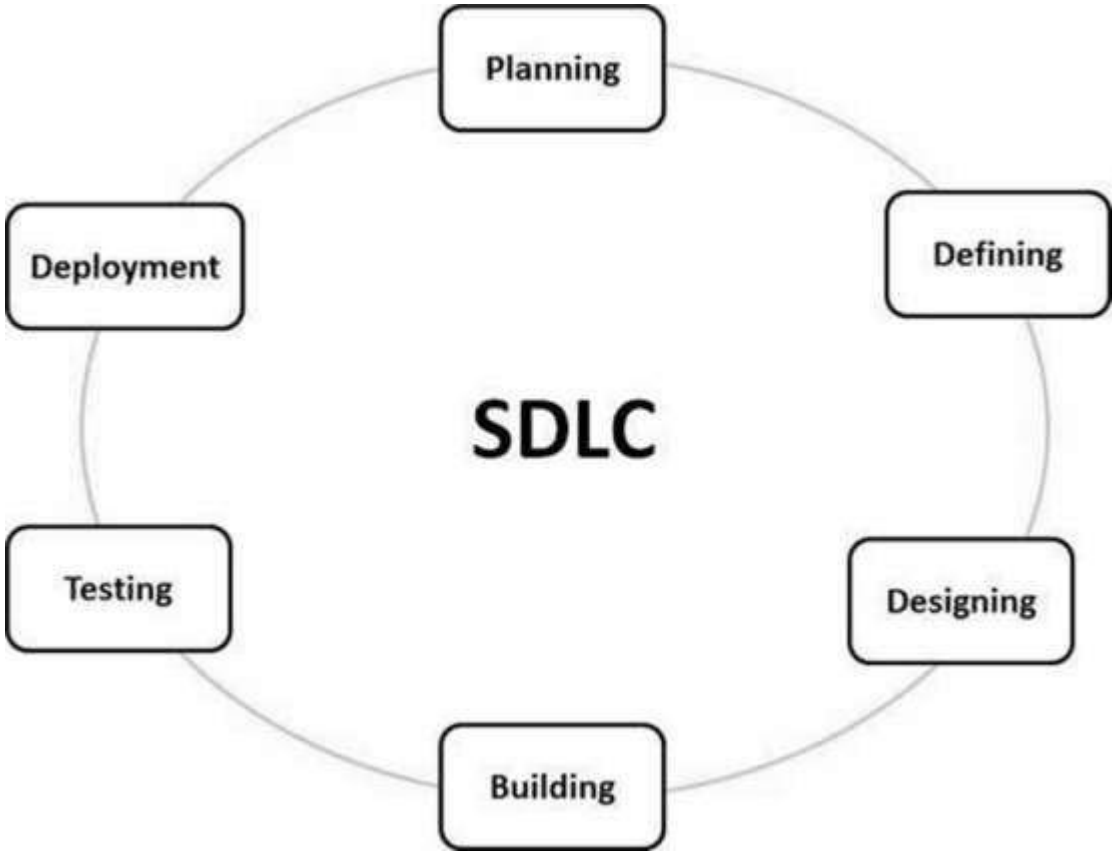

Fig 2. Steps in SDLC - adapted from [16]

Steps in the Software Development Life Cycle:

1. **Requirement Analysis and Planning -** Analysis of all the requirements is the most important and fundamental element in SDLC. Team members at senior level with feedback from customers, market surveys, sales department, and domain professionals are responsible for it. In this context, the main strategic project is outlined and a product feasibility study is carried out in the economic, operational and technological areas. Quality assurance demands and the risk identifying for the project are also planned in the planning phase. The technical feasibility assessment results in defining several technical methods to the effective execution of a low-risk project.

2. **Defining Requirements -** When the requirements analyses have been done, the following step is to explicitly describe and document product demands and to get their approval from customers and market analysers. The document includes all product needs that need to be identified and generated throughout the whole project life cycle.

3. **Product Architecture -** The Specification of Software Requirements (SRS) is the basis for the ideal architecture for the product for product architects. Based on the SRS, more than one product architectural design technique is generally proposed in a design document specification (DDS). Different features such as risk assessment, product robustness, modular design, budget and time limits are used to determine the ideal concept approach for the product.

   With their communication and data flow representations with external and third-party modules, all the architectural components are explicitly stated (if any). The internal design of all modules in the proposed architecture with the most precise information in DDS should be given clearly.

4. **Product Development -** The real development begins at this stage of SDLC and the product is produced. DDS is developing the programming code at this stage. The code may be developed without much difficulty if the design is done precisely and deliberately

   Developers must meet their organisational coding standards to generate their code; programming tools, such as compilers, translators, debuggers etc. are used. Diverse high level languages like C, Python, C++, Golang, Javascript, Java are utilised for encoding. The language is chosen based on the type of product.

5. **Testing Product -** This phase is usually an important portion of every phase as the test activity in the most recent SDLC models is generally carried out in all SDLC phases. However, the only test step of the product is reporting, monitoring, re-tested and tracking product faults until the product fulfils the quality requirements as stated in the SRS.

6. **Maintenance or Market Deployment -** Once tested and ready for deployment, the product is officially released. Product rollout occurs occasionally in stages in accordance with the business plan of the firm. In a tiny industry in the real business world the product can first be released and assessed (UAT- User acceptance testing).

   This input will then enable the product as it stands or with requested enhancements to be made public in the market segment. When put on the market, the product will be maintained for the current customer base.

2.1.2 Frontend Development

Development on the front handles everything in your browser or application that consumers see instantly. A front-end developer is responsible for the design and look of a website. The focus is on 'consumer side.' Front end developers will analyse code, build and debug apps and deliver user experience in a seamless way. You first manage the information you view in your browser. Your role as a front end developer is the look, feel and design of this website. HTML, CSS and front-end Javascript included. JQuery extends beyond that, such that many old apps still use the JavaScript library, which is why modern browsers can achieve the same work today, but much quicker than jQuery. There will also be learnt tonnes about reactive design, typography, arrangements, grid system and colour theory. When you plan on building websites and redeveloping them as a developer of the front end. You don't need to know how to construct the back end to be an early developer (often termed a Javascript developer). Developers' sites in front of the database will not interact with any working information.

2.1.3 Backend Development

The development background refers to the server side of the programme and the whole between the database and the browser. Returns development refers to the server on which you mostly focus on the functioning of the site. You will be primarily responsible for making updates and modifications as well as monitoring the website operation. Usually, this form of web development has three parts: a server, an app and a database. Backend developer-based code is the information that transmits the database to the browser. Either databases or servers is the job of a backend developer. Anyone who cannot see readily with the eye is working. Many developers of backend know the frontend languages like CSS and HTML, but employ languages like Python, C++, Ruby and Net to perform background work. Backend developers mostly focus on reactivity and speed of a site. These languages are used to develop dynamic websites that are distinct from static websites by saving information in these sorts of websites. The site content changes and updates regularly. E.g. Twitter, Google, Facebook and Google Maps are dynamic sites for example.

2.1.4 MongoDB

MongoDB is a document-oriented database source-available cross-platform. MongoDB uses optional patterned JSON-like documents categorised as a NoSQL database programme. For the Public Server License of MongoDB Inc., MongoDB has been built (SSPL). MongoDB is a database of documents that store data in JSON-like documents. I think this to be the most clear and expressive way of taking data into account than the standard line/column model. It's significantly stronger. To link Flask APIs to Mongo I used the Pymongo library.

Overview of MongoDB

- **Database:** Database is a collection of physical containers. Each database is provided with a file system with its own collection of files. Typically, one MongoDB server contains many databases.

- **Collection:** A group of MongoDB documents is collected. Collection The RDBMS table is equivalent. In a single database a collection exists. No schema is applied to collections. Documents can have several fields inside a collection. All documents in a collection usually have a similar or related function.

- **Document:** A group of MongoDB documents is collected. Collection The RDBMS table is equivalent. In a single database a collection exists. No schema is applied to collections. Documents can have several fields inside a collection. All documents in a collection usually have a similar or related function.

Table 1. Relation between MongoDB vs Relational DB

MongoDB	**Relational DB**
Database	Database
Document	Table
Collection	Row
Field	Column
Primary Key	Primary Key
Embedded Documents	Table Join

Main Features of MongoDB

MongoDB is a flexible and scalable NOSQL database solution for documents that aims to transcend the relational database approach and the restrictions of previous NoSQL alternatives. MongoDB provides an exceptional degree of scalability and flexibility, due to its horizontal scale-up load balance features.

MongoDB Atlas is the world's leading cloud database service. Developers may use Atlas to instal fully managed AWS, Azure or Google Cloud databases. Experienced practises of data

security and privacy standards allow developers to easily be aware of the immediate access they need to the scalability, availability, and compliance for the creation of company-level applications.

Top 5 features for technical use are:

- **Ad-hoc queries for optimal analytical data in real time**
    - It is difficult to know all queries done by end users while developing the scheme of a database. A short-lived ad hoc query is a variable-specific value command. The results may vary, based on the factors in question, whenever an ad hoc query is conducted.
    - Optimization of the way adhoc queries are handled can make a substantial impact on the basis of thousands of millions of variables. This is why MongoDB, which is a document-oriented and flexible database, is the cloud database option for companies requiring analytics in real time. The speed increase might change with query capability that enables developers to adjust the requests in real time. This cannot be achieved using other relational databases.
    - MongoDB enables regular expressions searching, range queries and field queries. Queries can take user-defined functions into account or return fields as required. This is done through MongoDB indexing and using the MongoDB query language (MQL) for BSON documents.

- **Indexing for enhanced query performance**
    - Experience has shown that the number one difficulty with many technical support teams is indexing. Done correctly, the search speed and performance is improved by indexes. If relevant indicators are not correctly defined, a number of accessibility difficulties, such as query execution and load balancing concerns, can and will generally lead to.

- A database is compelled to scan documents one by one without the necessary indexes to identify those which match the query expression. However, if there is a corresponding index for each query, the server may optimise the user requests. MongoDB offers a wide range of indices and capabilities that facilitate sophisticated access patterns to datasets with language-specific sort commands. Moreover it is better.
- In order to adapt in real time, constantly changing query patterns and application needs, MongoDB indexes may be generated on demand. You may also specify them in any field of any document, even those in arrays.

- **Data replication for better stability and availability**

  - If only one database contains your data, various possible faults like a service disruption, whole server crash or might be just a hardware failure can be experienced. Any such catastrophe would make it almost hard to retrieve your data. But this is easier with MongoDB.
  - Replication enables you to overcome these weaknesses by using many servers to recover and backup disasters. A horizontal scaling of the same information (or data shares) over numerous servers means that data availability and stability are boosted significantly. Replication of course also helps to balance loads. The burden can be evenly divided on servers if several users access the same data. Time taken would be less.
  - Replica sets for this reason are used in MongoDB. A primary server or node takes all written actions and replicates the data through subsidiary servers. If the primary server ever has a major breakup, a new primary node can be chosen by any of the subordinate servers. And if the previous primary node returns online, the new primary node is a secondary server.

- **Sharding**

    - When dealing with huge data settings, it allows the data base to spread and more effectively execute what may be difficult and lengthy queries if it is divided by enormous datasets into numerous dispersed collections and 'shards.' Without sharding, it is almost impossible to scale a growing Web App with millions of daily visitors.
    - Shorting MongoDB offers significantly more horizontal scalability, just like replication using replication sets. Horizontal scaling means that a part of the data set in each cluster is mostly used as a separate database. A single, complete database, which can handle the demands of popular, expanding applications with no downtime, is the collection of dispersed server shards.
    - Every activity in a sharding system is managed by a lightweight process named mongos. Based on the shard key, Mongos may route requests to the right shard. Of course, appropriate sharding also helps to improve the load balancing.

- **Load balancing**

    - Load balancing is always required at the backend. Properly spreading billions of requests / API calls from clients to thousands of servers may lead to speed increase that is obvious (and welcomed).
    - Fortunately, MongoDB offers large-scale load balancing with horizontal scaling capabilities such as replication and sharding. The platform can manage a number of simultaneous read and write requests with the most advanced competitiveness control and data locking methods for the same data. An external load balancer does not need to be added—MongoDB guarantees that each user gets a consistent view of the data they need to access.

```
{
   _id: ObjectId(7df78ad8902c)
   title: 'MongoDB Overview',
   description: 'MongoDB is no sql database',
   by: 'tutorials point',
   url: 'http://www.tutorialspoint.com',
   tags: ['mongodb', 'database', 'NoSQL'],
   likes: 100,
   comments: [
      {
         user:'user1',
         message: 'My first comment',
         dateCreated: new Date(2011,1,20,2,15),
         like: 0
      },
      {
         user:'user2',
         message: 'My second comments',
         dateCreated: new Date(2011,1,25,7,45),
         like: 5
      }
   ]
}
```

Fig 3. Sample document in MongoDB

2.2 Backend Development on Python

It is obvious that it makes it simpler for Python to participate in current planning techniques. Python contains several potent libraries with an enormous quantity of pre-written code. Developers don't have to write code from scratch so that the development time is speeded up. This makes using Python for backend development a great alternative. There are a couple of causes for the background of enterprises or organisations in python.

- **Object-oriented programming:** As we know, the most significant element for programming is object-oriented capability. Python is a computer language created primarily for different object-oriented concepts. In addition, for developers, it is the primary choice. The most interesting element of this is that PHP 5 has lately been

published as a programme support for objects. This is one of the reasons why programmers embrace Python object-oriented programming. Various actions and characteristics are arranged into many objects with object-oriented programming. Everybody has his/her own role. If an error happens in any portion of the code, object-oriented programming does not affect the remainder of the code.

- **Easy to comprehend the code:** Python's readability is the primary advantage of designing backend applications. His readability is mostly due to the usage of whitespace to outline the code units. It helps you evade the thick character woods. Apparently, Python does not employ semicolons and second brackets. Just hit the input key to terminate the line. As the code is easy to read, it will not be difficult for developers to comprehend and to grasp the code the coder has produced. Consequently, the development time is greatly reduced.

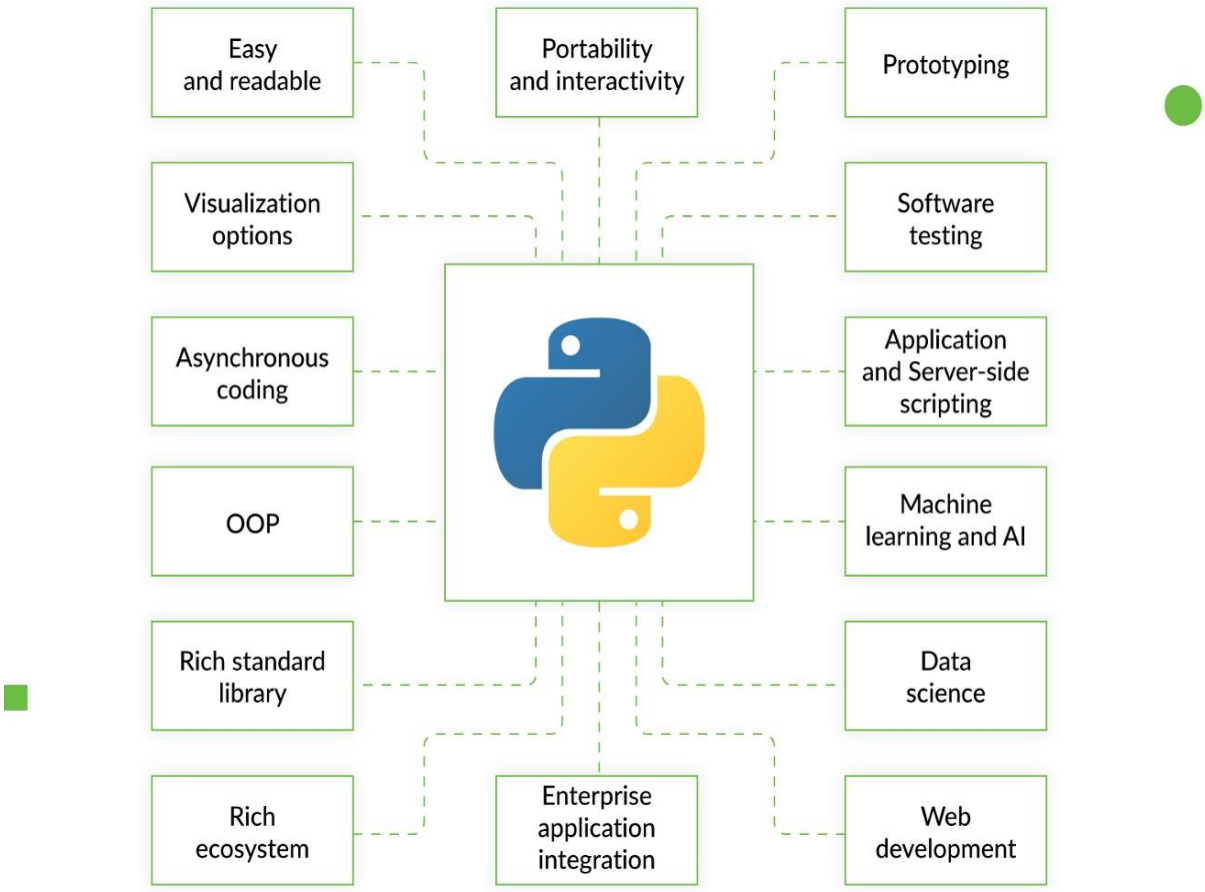

Fig 4. Advantages of using Python for Backend Development - adapted from [17].

2.2.1 Flask framework

Flask is a micro web framework built in Python. It is characterised as a micro-framework since it requires no tools or libraries in particular. It doesn't have the abstraction of the database, a form validation or any other component that has a common function of current third-party libraries. However, Flak offers extensions that can add application characteristics in a way that Flask itself implements. Extensions exist for form validation, upload processing, numerous standard utilities and object-relational mappers, connected to the open authentication technology.

LinkedIn, Pinterest, etc. are feature applications using the Flask framework.

Components of Flask:

- **Werkzeug** : Werkzeug, in other words, is a toolkit for the applications of web server gateway interface (wSGI) and is published under a BSD licence. It is also a toolkit for Python programming language programming language. For request, response and utility operations, the tool may create software objects. It supports Python 2.7 and 3.5 later, it is used to construct a bespoke software framework above it.

- **Jinja :** Jinja is a Python programming language web template engine. It is designed and released under a BSD License by Armin Ronacher. Jinja resembles the Django template but gives Python-like terms while making sure the templates are assessed by the sandbox. It is a template language based on text and can thus be used for generating any benchmark and source code. The Jinja model enables tags, filters, tests and global customisation. Jinja also permits the designer to invoke functions with parameters about objects, as opposed to the Django template motor. Jinja is the default template motor for Flask.

- **ItsDangerous :** ItsDangerous is a secure Python programming language data serialisation package distributed under a BSD licence. It is used to save the Flask session in a cookie, without enabling users to manipulate the session content.

- **MarkupSafe :** MarkupSafe is a string handling library provided under BSD licence to the Python programming language. MarkupSafe extends the type of Python string to designate its contents as "safe;" combined with ordinary strings the unmarked strings automatically escape without duplicating the already marked strings.

Features of Flask:

- Server and debugger development
- Integrated unit test support
- Uses Jinja templates RESTful request dispatch
- Safe cookie support (client side sessions)
- 100% WSGI 1.0 Unicode-based compliance
- Large documentation
- Compatibility of Google App Engine
- Extensions to improve the intended features

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello() -> str:
    return "Hello World"


if __name__ == "__main__":
    app.run(debug=False)
```

Fig 5. Simple Flask Application

2.2.2 Hadoop Distributed File System (HDFS)

HDFS is the number one data warehouse gadget which is used in Hadoop programs. Its has components which are namely NameNode and DataNode structure to put in force a disbursed document gadget that provides excessive-performance to get entry to statistics across extraordinarily scalable Hadoop clusters.

HDFS presents a reliable way for handling pools of large information and assisting associated big data analytics applications.

HDFS makes use of a architecture which is similar of a masternode as theyll as slavenode. In its preliminary formation, each A.H cluster consists of a standalone NamingNode that controlled report system operations and helping DataNodes that managed information garages on man or woman compute nodes

2.2.3 Advantages of HDFS

- Scalable

  Hadoop is a noticeably scalable storage platform because it is able to get masses of inexpensive servers and use parallel processing. Unlike conventional relational database structures (RDBMS) which can’t scale to technique, massive amounts of facts.

- Cost-powerful

  Hadoop also gives a value-powerful storage answer for groups exploding facts units. The trouble with conventional relational database management systems is that it's extremely value prohibitive to scale to this kind of degree, a good way to use such large volumes of facts. In an attempt to lessen fees, many groups inside the past might have needed to down-sample information and classify it based totally on certain assumptions as to which records become the maximum value. The raw records might be deleted, as it'd be too cost-prohibitive to keep.

- Flexible

  Hadoop permits businesses to easily get admission to new information assets and tap into distinct styles of statistics to generate value from those statistics. This way in which any type of organization can use Hadoop to generate valuable commercial enterprise insights from data sources along with social media, electronic mail conversations.

- Fast

  Hadoop's specific garage approach is primarily based on a disbursed file gadget that basically 'maps' facts anywhere it's far positioned on a cluster. The gear for facts processing is regularly on the equal servers in which the facts are located, resulting in a lot faster records processing. If you're handling large volumes of unstructured facts, Hadoop is capable of successfully technique terabytes of data in just mins, and petabytes in hours.

- Resilient to failure

  Fault tolerance is one of the advantages. When files are sent to a person node, that information is likewise replicated to different nodes in the cluster, this means that that within the occasion of failure, there is every other copy to be had for use.

2.3 Cloud Computing and Architecture

The transportation of on-demand computer services is cloud computing — from packs to garages and processing strength. PC machine assets, in particular data garage (cloud storage) and computer energy, are available on demand, without direct energetic person supervision. The time-frame is frequently utilised to provide information on the Internet for numerous clients. Large clouds are generally assigned capacity from many main servers, predominantly these days.

The following are the types of cloud categories:

- Infrastructure-as-a-Service: IAaS refers to the core computer building pieces that may be leased. Infrastructure as a Service: bodily or digital servers, storage and networking. This is appealing to agencies that need to construct packages from the very ground up and need to manipulate nearly all of the factors themselves, however, it does require corporations to have the technical skills so one can orchestrate services to that degree.

- Platform as a Service: PaaS will include a setup of middleware programmes, database controls, working structures and development tools — aside from the underlying storages, networking and digital servers, this will also include the gear and software programme developers must build programmes on top of.

- Software-as-a-Service: SaaS is defined as the delivering of the packages-as-a-service, in all likelihood the model computing on the cloud's environment for the general public on an everyday basis. The hardware running system is beside the point to the stop consumer, who will get admission to the provider via a web platform or applications. The miles frequently brought on an in step with-seat or according-to-user basis.

2.1 Summary of Research Papers Studied

The research papers that are studied to carry out the work in the field of Histopathology and Semi Supervised Learning.

Title: Introduction to Semi-Supervised Learning [1]

The authors of this book tell us everything about the Semi-supervised learning. It is a learning paradigm that aims to examine how computers and natural systems like human beings learn using labelled and unlabeled input. Traditionally, learning has either been investigated in the unattended paradigm (e.g. clustering, outlining) where all the data is unlabeled, or in a monitored (e.g. classification, regression) paradigm with all the data being labelled. The aim of semi-monitored learning is to understand how the mix of labelled and unscheduled input might influence the learning behaviour and create algorithms that benefit from a mixture of these. Semi-supervised learning is of considerable interest in machine learning and data mining, because unlabeled data may easily be used to enhance supervised learning tasks if the information on the labels is rare or costly. Semi-supervised training is also a quantitative technique for understanding learning in the human category, where the majority of the input is obviously unlabeled. They discuss some of the common half-monitored learning models including self-learning, combination models, co-training and multi-view learning, graphic methodology and half-vector support machines in this introductory book. They describe their essential mathematical wording for each model. Certain fundamental assumptions are important to the effectiveness of semi-supervised learning. They underline every model's assumptions and offer counter-examples to show the limits of the different models wherever applicable. Moreover, semi-monitored cognitive psychology education is discussed. In conclusion, they offer a computer-supervised learning theoretical perspective and end the book with a short discussion of open-ended problems on the ground.

Title: Adaptive Semi Supervised Support Vector Machine Semi Supervised Learning with Features Cooperation for Breast Cancer Classification [2]

Semi-supervised education is the machine learning branch that carries out specific learning activities by employing both labelled and unlabeled data. It is designed to allow the use of huge volumes of unscreened data that are accessible in many circumstances together with generally smaller quantities of labelled data, between supervised and non-screened training. Research has followed the general trends observed in machine learning in recent years with a great deal of emphasis paid to neural network models and generative learning. The topic literature also grew to cover a wide variety of theories, methods and applications in volume and breadth. Due to the exponential rise of conducted mammograms, computerised diagnosis (CAD) of cancer of the breast is becoming a need. The diagnosis and categorization of breast mass, in particular, is presently of major concern. The second major cancer cause (post-lung cancer) deaths in women and survival rates are substantially influenced by early stage discovery.One of the most prevalent kinds of cancer is breast cancer. In order to reach a final judgement, the CAD systems are built on three primary steps. The vast amount of the acquisition data that must be labelled in a precise manner is normally characterised by CAD systems. However, collecting patient records is not straightforward. A patient's record is 'survived' or 'non-survived' for at least five years. What leads to a serious concern is the need for an expert to perform the labelling. That is why, using semi-supervised training, the statistical learning community has sought to satisfy these practical demands (SSL). For that reason they have presented a diagnostic CAD system based on a special method of applying the semi-supervised learning methodology by employing S3VM with these features. We've made many empirical decisions.

Title: Imagenet: A large-scale hierarchical image database [3]

The explosion of picture data on the Internet can help to index, retrieve, organise and interact with pictures and multimedia databases with more advanced and storable models and algorithms. However, it remains a crucial challenge just how such data might be used and arranged. Here, you will present a new database named "ImageNet," an extensive ontology of pictures built on the WordNet's backbone. ImageNet seeks to supplement most of WordNet's 80,000 synsets with an average of 500-1000 clean pictures in full quality. This results in the creation, through semantic hierarchy of WordNet, of tens of million annotated frames. This study provides a comprehensive examination of ImageNet as it stands: 12 subarms with 5247 synsets and a total of 3.2 million pictures. They show that ImageNet is substantially bigger and more diverse than the present picture databases. It is a difficult effort to construct such a vast database. They discuss Amazon Mechanical Turk's data collecting technique. Finally, they show the efficacy of ImageNet by means of three basic object identification, imaging and autonomous object grouping applications. They anticipate ImageNet will provide scholars unique chances in the computer with its scale, precision, variety and hierarchical structure

Title: 1399 H&E-stained sentinel lymph node sections of breast cancer patients: the CAMELYON dataset [4]

One of the most significant criteria in the prognosis of breast cancer is the existence of lymph gland metastases. The lymph gland method is one of the most popular means of evaluating the regional status of the lymph gland. The lymph gland sentinel is the most common lymph gland containing metastatic cells of cancer and is removed by a pathologist, histopathologically processed and analysed. This time consuming arduous test approach might lead to the missing of tiny metastases. Recent progress has, however, provided a road to examination of scanned parts with computer algorithms using entire slide imaging and machine learning. The data collection was compiled in 3 TB of data for the grand challenges of CAMELYON17 and CAMELYON16 These challenges, comprising 1399 whole-slide images (WSIs) with annotations of the lymph glands, with metastases or without metastases. Five medical facilities gathered slides with a wide variety of imaging and staining variables. Every WSI has a diaphragm label that indicates if it does not include metastasizes, macro-metastases, micro-metastases, or isolated tumour cells. In addition, precise outlines created by hand are supplied for all metastases for 209 WSIs. Lastly, open source software tools have been made accessible for viewing and interacting with the data. The great potential for reuse has been supplied with a unique data collection of annotated, complete digital histopathology photographs.

CHAPTER 3

SYSTEM DEVELOPMENT

This chapter contains the overall description of the project, different requirements for the project.

3.1 Overall description of the project

During my internship period (ongoing) I was part of 2 projects. Initially when I joined Nference I was part of the Easy Augmentation project on which I mainly worked on Backend Development. After I was a part of Patient Explorer app where mostly fronted development was required.

3.1.1 Easy Augmentation

Easy Augmentation is a Project which yield us result of model run. It is initially fed with certain cohort and machine learning models are run on them which gives us confidence score for each patient against certain disease synonym.

3.1.2 Patient Explorer

In this project there was an existing UI which got redesigned, and I was given task create it . Patient Explorer is an app which shows all Patient notes of patients and timeline of all their medical history. Patient can be an individual patient or can be part of a cohort.

3.2 Requirement Analysis

This section contains the hardware and software requirements to carry out the project.

3.2.1 Hardware Requirements

Development Environment:

Only CPU Server:

- RAM: 64 GB
- CPU Cores: 16 cores
- DiskSpace: 1 TB
- OS: Linux Ubuntu 18.04 x86
- Location: Canada

Production Environment:

GPU Server:

- RAM: 16 GB
- CPU Cores: 16 cores
- DiskSpace: 1 TB
- GPU: 16 GB VRAM
- OS: Linux Ubuntu 18.04 x86
- Location: Germany

CPU Server:

- RAM: 64 GB
- CPU Cores: 16 cores
- DiskSpace: 1 TB
- OS: Linux Ubuntu 18.04 x86
- Location: Canada

3.2.2 Software Requirement

Operating System - Linux Ubuntu 18.04 x86

Python (>= 3.6)

NGinx Web Server

MongoDB

SMTP Server

Python Libraries -

- import traceback
- import os
- import json
- from concurrent.futures import ThreadPoolExecutor
- from pymongo import MongoClient
- import os
- import hashlib
- from flask import Flask, jsonify, request
- import pymongo
- import numpy as np
- import pandas as pd

CHAPTER 4

PERFORMANCE ANALYSIS

4.1 Easy Augmentation

Easy Augmentation is a Project which yield us result of model run. It is initially fed with certain cohort and machine learning models are run on them which gives us confidence score for each patient against certain disease synonym.

4.1.1 Data Used-

Most of the data is data from Mayo clinic. Researchers can draw valuable scientific and clinical insights using the Clinical Data Analytics Platform, which includes best-in-class Nference deidentification technology, augmented curation of unstructured and structured data, and a revolutionary federated search model. The data is kept in a safe cloud environment. This ground-breaking platform prioritises patient privacy while offering up exciting new research and development opportunities in medicines, early diagnosis, and clinical care.

4.1.2 Workflow

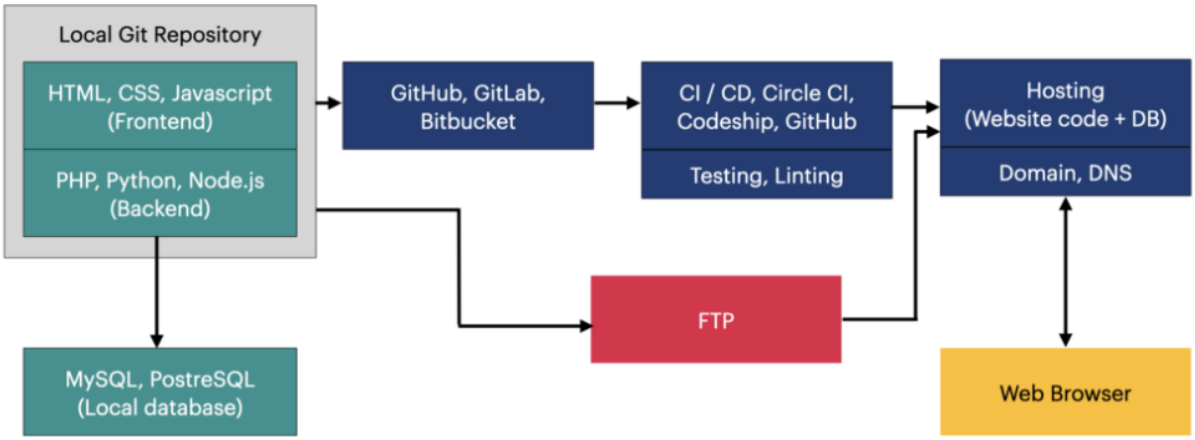

Fig 6. Workflow of web development

Celery Task-

Celery is a free, open-source Python library for asynchronous task execution. It's a task queue that collects and distributes tasks to workers in a timely manner. It is primarily designed for real-time use, although it also allows for scheduling (run regular interval tasks). It significantly improves the end user activities. Celery is a free, open-source Python library for asynchronous task execution. It's a task queue that collects and distributes tasks to workers in a timely manner. It is primarily designed for real-time use, although it also allows for scheduling (run regular interval tasks). It significantly improves the end-user's.

When we send a request to the server through the client in the standard HTTP request-response cycle, the server responds to the client. It works well for minor jobs, but when we try to load massive tasks, it may become slow. As a result, we'll need to create a feature that reduces load time.

Celery communicates by messages; typically, a broker acts as a middleman between clients and employees. Celery's inner workings confirm a pattern of Producer and Consumer. Celery possesses all three primary elements at a high level.

Producers - Producers are the "web nodes" that handle web requests. Tasks are allocated to the Celery means forced into the task queue while the application is processing.

Consumers - Consumers are the 'worker nodes' that watch the queue head, and workers take on and complete the jobs. Workers can also do a variety of jobs, allowing them to act as producers.

API usage and sharing

The APIs are shared with the team mates using **Postman.** This application is a collaborative platform for API development. We can also test the results of the API here.

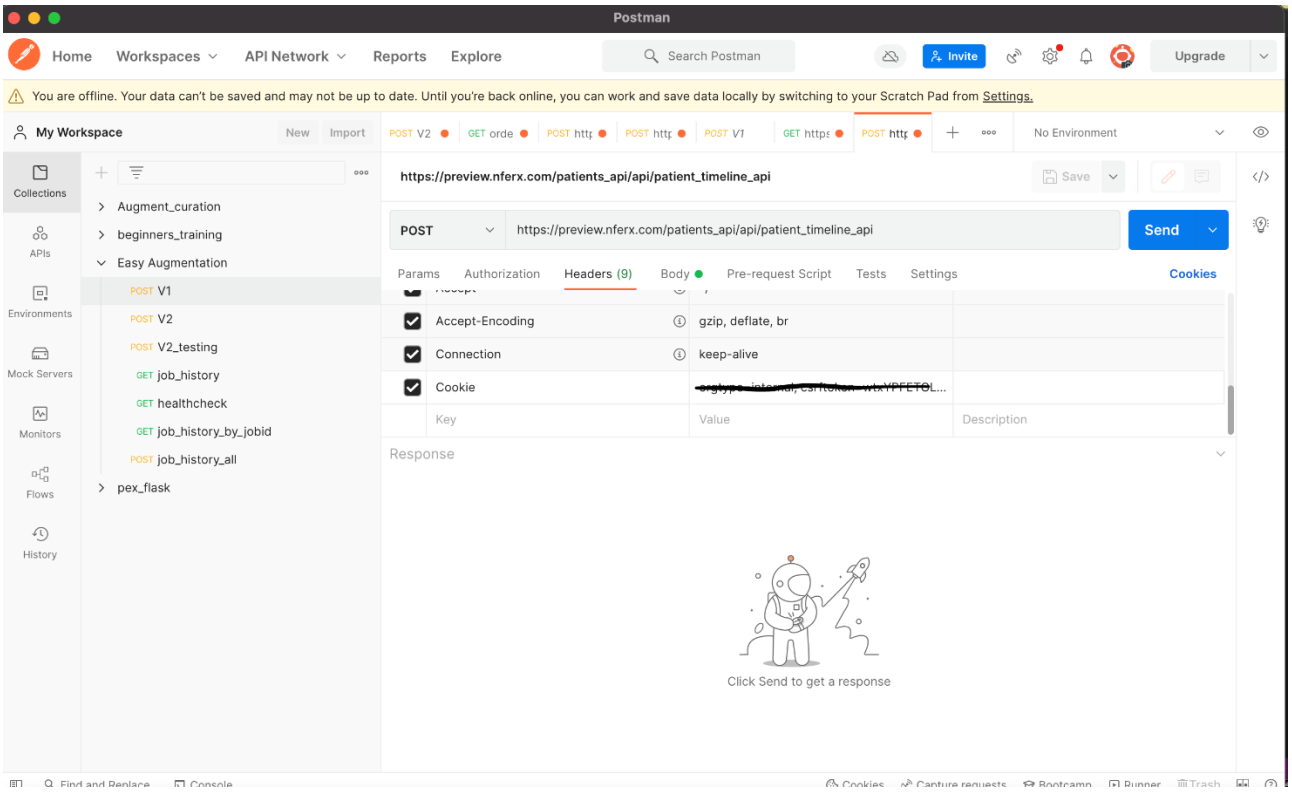

Fig 7. Postman

4.1.1 Database Used - MongoDB

MongoDB is cross-platform software for the document-oriented source database. MongoDB uses optional patterned JSON-like documents classed as a NoSQL Database Program. MongoDB is designed by MongoDB Inc. for the Public Server Side License (SSPL). MongoDB is a database document that keeps information in documents such as JSON. We feel that this is the most clear and expressive technique to taking data into consideration as the conventional model row/column. It is significantly stronger. To link Flask APISs to Mongo I used the Pymongo library.

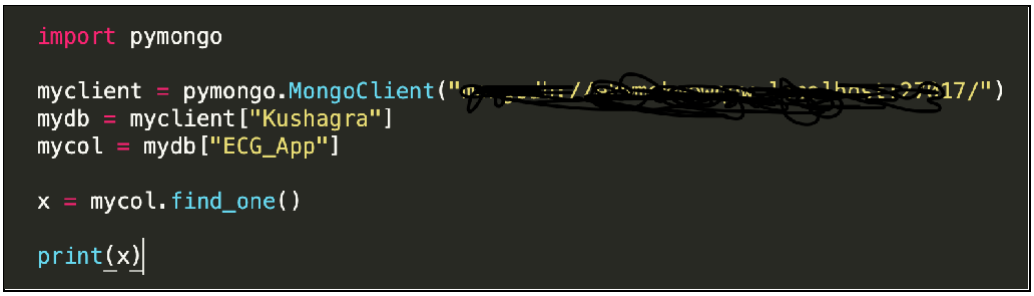

Fig 8. Simple Flask Application

4.2 Patient explorer

In this project there was an existing UI which got redesigned, and I was given task create it . Patient Explorer is an app which shows all Patient notes of patients and timeline of all their medical history. Patient can be an individual patient or can be part of a cohort.

4.2.1- High Level Design

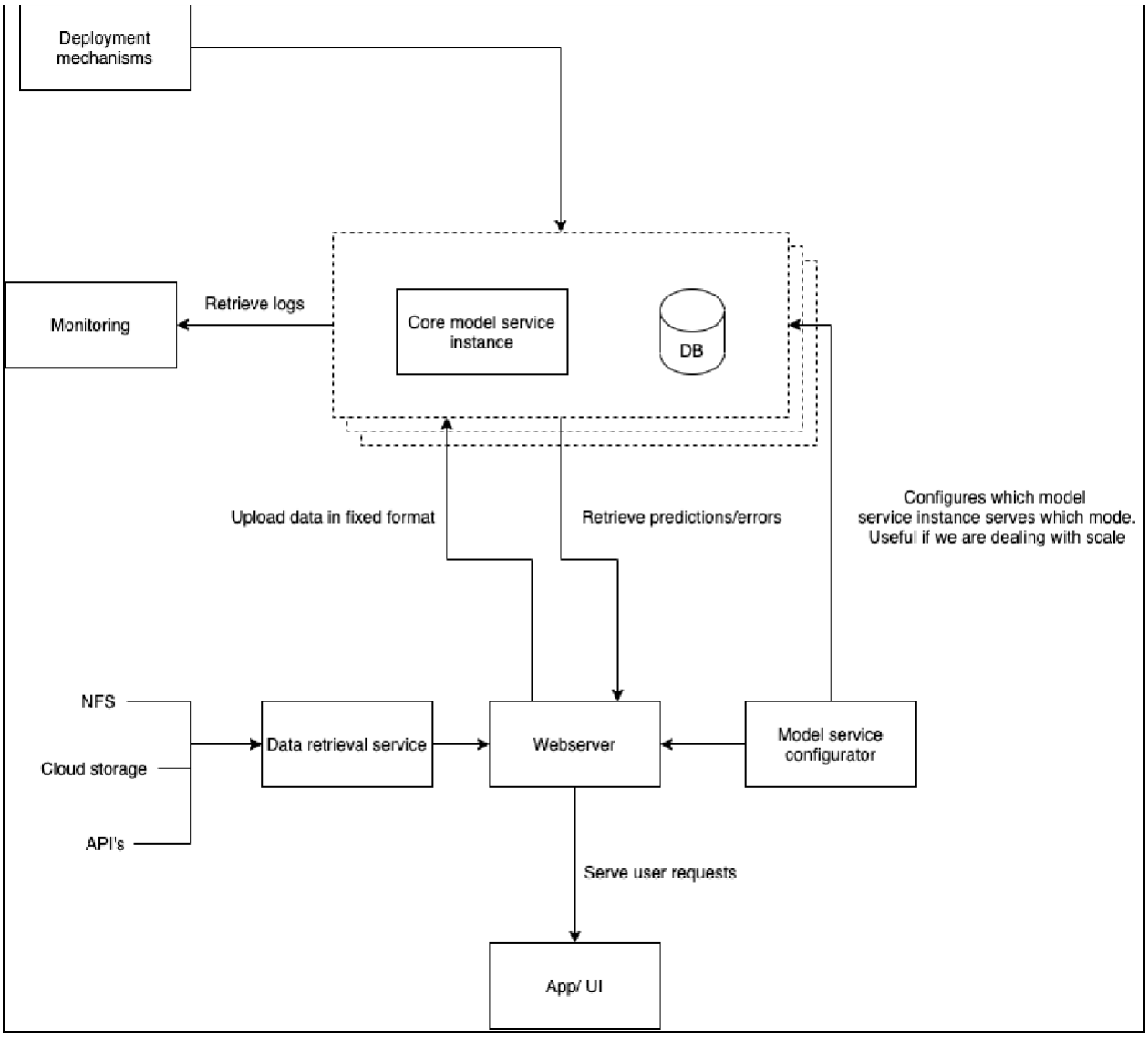

Fig 9. High level design of app

4.2.2 React

React.js is an open-source JavaScript package that is used to create single-page apps' user interfaces. For web and mobile apps, it's utilised to manage the view layer. We can also make reusable UI components with React.

Developers can use React to build massive web applications that can alter data without reloading the page. React's major goal is to be quick, scalable, and easy to use. It only works on the application's user interfaces. This relates to the MVC template's view. It can be combined with other JavaScript libraries or frameworks, such as Angular JS in MVC applications.

The render() method in React components takes input data and returns what should be displayed. This example use the JSX syntax, which is similar to XML. This is how render() gets access to input data supplied into the component. props.

4.2.3 Material UI

MUI is a huge library of UI components that designers and developers can use to create React apps. The open-source project adheres to Google's component-creation rules, providing you with a customisable collection of basic and complex UI elements.

MUI also sells a set of React templates and tools, allowing you to customise pre-made user interfaces for your project.

4.2.4 Nference Core UI components

Nference has its own predefined components which were extensively used in the project.

4.2.5 CSS

CSS, or Cascading Style Sheets, is a simple design language designed to make the process of making web pages presentable easier.

The style and feel of a web page is handled by CSS. You can use CSS to manage the colour of the text, font style, paragraph spacing, how columns are scaled and laid out, what background pictures or colours are used, layout designs, display variants for different devices and screen sizes, and a multitude of other effects.

CSS is simple to learn and understand, but it gives you a lot of power over how an HTML document looks. CSS is frequently used in conjunction with the markup languages HTML or XHTML.

4.2.6 HTTP Methods

POST, GET, PUT, PATCH, and DELETE are the most regularly used HTTP methods. These methods relate to the CRUD (create, read, update, and delete) actions. There are a few more options as well, but they're used less commonly.

HTTP Method	CRUD operation	Entire Collection (e.g. /users)	Specific Item (e.g. /users/{id})
GET	Read	200 (OK), list of entities. Use pagination, sorting and filtering to navigate big lists.	200 (OK), single entity. 404 (Not Found), if ID not found or invalid.
POST	Create	201 (Created), Response contains response similar to **GET** /user/{id} containing new ID.	**not applicable**
PATCH	Update	Batch API	200 (OK) or 204 (No Content). 404 (Not Found), if ID not found or invalid.
DELETE	Delete	204 (No Content). 400(Bad Request) if no filter is specified.	204 (No Content). 404 (Not Found), if ID not found or invalid.
PUT	Update/Replace	**not implemented**	**not implemented**

Fig 10. HTTP Methods

4.3 TESTING (QUALITY OF ROBUSTNESS)

In this chapter I'll talk about the different types of testing / benchmarking that has been done for the quality of the robustness of the product.

4.3.1 Testing Plan

Table 2. Testing Plan

Type of Test	**Will The Test Be Performed?**	**Explanations**	**Software Component**
Requirement Testing	Yes	To check the feasibility of our project in terms of budget, requirements.	Not Attempted
Unit	Yes	Individual units of source code will be run with operating procedures to see if they are fit to use	Flask
Integration	Yes	The built test cases and test data are integrated and then predictions are made. The bugs (if found)	Python

Performance	Yes	To test whether our project will work well under expected workload, this is a must.	
Security	Yes	As the software willbe used in Clinics, the samples should be encrypted.	Python Encryption
Compliance	No	Not required	Not Attempted
Load	No	Enough Memory	Not Attempted
Volume	No	Not required	Not Attempted

Component Decomposition and Type of Testing Required

Table 3. Component Testing

S. No.	List of various functions that require testing	Type of testing required	Technique for writing cases
1.	Validation Accuracy	Integration	White box
2.	Flask API	Integration	White Box
3.	System Testing	System	Black box

4.3.2 Test Environment

GPU Server:

- RAM: 16 GB
- CPU Cores: 16 cores
- DiskSpace: 1 TB
- GPU: 16 GB VRAM
- OS: Linux Ubuntu 18.04 x86
- Location: Germany

CPU Server:

- RAM: 64 GB
- CPU Cores: 16 cores
- DiskSpace: 1 TB
- OS: Linux Ubuntu 18.04 x86
- Location: Canada

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion and Results

The following results / conclusions can be made from the project:

- Easy augmentation project works well within time. The time taken for a job to finish is minimal.
- The job after is completed, the user is notified by a mail.
- All the job results are stored in vormir for where they can looked into.

- In Patient explorer app, User can select either from a Patient or cohort and look for any patient note. Many filters can be applied to get the desired result.

- User can patient timeline of his/ her lab record, notes etc.

5.2 Future Scope

For the easy augmentation project when now any patient id fails we simply ignore those and give results for the rest. But in future we would need to store those patient id and after sometime rerun our job again for them.

Improvement in Timeline of Patient notes. They are just in a tabular Format and with only high level of information. In the future deep details might be required for scientist to analyse data carefully .

References

[1] Adaptive Semi Supervised Support Vector Machine Semi Supervised Learning with Features Cooperation for Breast Cancer Classification - Zemmal, Natheyl, Azizi, Nabiha, Dey, Nilanja, Sellami, Mokhtar https://www.ingentaconnect.com/contentone/asp/jmihi/2016/00000006/00000001/art00006

[2] D. J. Reifer, "Web development: estimating quick-to-market software," in *IEEE Software*, vol. 17, no. 6, pp. 57-64, Nov.-Dec. 2000, doi: 10.1109/52.895169.Vol. 3, No. 1 , Pages 123-130 - https://doi.org/10.2200/S00196ED1V01Y200906AIM006

[3] C. Barry and M. Lang, "A survey of multimedia and Web development techniques and methodology usage," in IEEE MultiMedia, vol. 8, no. 2, pp. 52-60, April-June 2001, doi: 10.1109/93.917971.

[4] Mandela Patrick, Yuki M. Asano, Polina Kuznetsova, Ruth Fong, João F. Henriques, Geoffrey Zweig, & Andrea Vedaldi. (2020). Multi-modal Self-Supervision from Generalized Data Transformations. https://arxiv.org/abs/2003.04298

[5] Yun Liu, Krishna Gadepalli, Mohammad Norouzi, George E. Dahl, Timo Kohlberger, Aleksey Boyko, Subhashini Venugopalan, Aleksei Timofeev, Philip Q. Nelson, Greg S. Corrado, Jason D. Hipp, Lily Peng, & Martin C. Stumpe. (2017). Detecting Cancer Metastases on Gigapixel Pathology Images.

[6] https://www.mongodb.com/https://www.tutorialspoint.com/sdlc/sdlc_overview.htm

[7] https://www.python.org/

[8] https://flask.palletsprojects.com/en/2.0.x/

[9] https://www.qualitasit.com/product-development/

[10] https://www.digitaltrends.com/cool-tech/dna-data-catalog-startup/

[11] https://www.kdnuggets.com/2020/08/top-10-lists-data-science.html

[12] https://www.edureka.co/blog/ai-vs-machine-learning-vs-deep-learning/