A

Project Report

On

Brain Tumor Classification

Submitted by

Fenil Ramani 2205101110085

as

Partial fulfilment of Semester VI

of Bachelors of Computer Applications

for A.Y. 2024-2025

Under the Guidance of

Internal Guide Name : Prof Vaishali Shah

Company name : Twilearn Edutech

Submitted To

Parul Institute of Computer Application,

Faculty of IT & Computer Science

Acknowledgement

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

I respect and thank Dr. Priya Swaminarayan, Dean, FITCS for providing us an opportunity to do the project work in BCA and giving us all support and guidance which made us complete the project duly. We are extremely thankful to Mam for providing her support and guidance, although she had a busy schedule managing the academic affairs.

We would not forget to remember Prof. Hina Chokshi, HOD, BCA department for her encouragement and more over for her timely support and guidance till the completion of our project work.

We owe our deep gratitude to our project guide Prof.Vaishali Shah, who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

I am thankful to and fortunate enough to get constant encouragement, support and guidance from our Parents, all Teaching staff of the BCA Department which helped us in successfully completing our project work. Also, we would like to extend our sincere esteems to all staff in the laboratory for their timely support.

Fenil K Ramani 2205101110085

PARUL INSTITUTE OF COMPUTER APPLICATION

CERTIFICATE

This is to certify that ***Fenil Ramani*** the students of Parul Institute of Computer Application, has/have satisfactorily completed the project entitled "Brain Tumor Classification**"** as a part of course curriculum in BCA / BSCIT semester- 6 for the academic year 2024-2025 under guidance of ***Prof.Vaishali Shah.***

Enrolment Number: 2205101110085

Quality of work	**Grade** **A / A+ /B /B+**	**Sign of Internal guide**
Poor / Average / Good /Excellent		

Date of submission:

HOD

Dr. Hina Chokshi

DEAN

Dr. Priya Swaminarayan

ABSTRACT

- Brain tumor classification and detection are crucial for accurate diagnosis and treatment planning. Traditional methods depend on radiologists analyzing MRI and CT scans, but recent advancements in artificial intelligence (AI) have improved these processes. Deep learning models, especially convolutional neural networks (CNNs), can automatically identify and classify brain tumors, enhancing diagnostic accuracy and efficiency.

- AI models excel in distinguishing between different tumor types, such as gliomas and meningiomas, by recognizing patterns in imaging data. Combining AI with multi-modal imaging and radiomics has further improved detection sensitivity and specificity.

- Despite these advances, challenges such as data imbalance and model generalizability persist. Future developments focus on enhancing AI models, reducing false positives, and making them more interpretable to assist clinicians in making better-informed decisions.

- However, challenges like data imbalance and the need for generalizable models remain. Future research aims to improve model performance, reduce false positives, and develop explainable AI systems to support clinical decision-making.

- Brain tumor classification is a critical task in medical diagnosis and treatment planning. Accurate identification of tumor types, such as benign or malignant, can significantly impact patient outcomes. With advancements in artificial intelligence, machine learning (ML) techniques have emerged as powerful tools for automating and enhancing the classification process.

- Evaluation metrics, including accuracy, precision, recall, and F1 score, are employed to compare model performances. Results demonstrate the potential of CNN-based models to achieve high classification accuracy, particularly when trained on large, diverse datasets.

INDEX

1. Company Profile

Twilearn Edutech (India) Pvt Ltd

Address: Bhumkar Das Gugre Rd, Bhagwan Nagar, Bhumkar Nagar, Wakad, Pimpri-Chinchwad, Maharashtra 411057

Overview :

TwiLearn Edutech Private Limited, In the wake of the digital revolution in India, Twilearn- Pay After Placement is an online learning platform that trains IT aspirants to land their dream jobs. The platform offers courses in domains aligned with the latest tech and market trends, providing opportunities for securing jobs with competitive salaries. Twilearn focuses on catering to the increasing demand for skilled IT professionals in India by training freshers and experienced learners online with live classes, case studies, hands-on exercises, and industry projects.

Twilearn is an innovative edtech platform designed to bridge the gap between education and the competitive job market. At Twilearn, we empower students and job seekers to gain real-world experience by working on industry-relevant projects. Through hands-on learning, they build practical skills that are highly valued by employers. Our platform offers a unique combination of project-based learning and internship opportunities, providing participants with a solid foundation in their chosen field. Along with valuable project exposure, users earn internship certifications that enhance their resumes and increase employability. Twilearn is committed to helping individuals gain the expertise they need to stand out in a competitive job market, offering mentorship, guidance, and real-world insights that lead to successful career placements. Whether you're a student looking to enhance your academic knowledge or a job seeker aiming to improve your professional profile, Twilearn is your gateway to a brighter future.

Contact Information: Address: 406 Atlantis Heights, Sarabhai Campus Near Genda Circle, Vadodara-390023, Gujarat, India

Email: hr@twilearn.in

Website: https://twilearn.in/

2.Project Profile

2.1 Project Definition

The Productivity Tool stands at the forefront of innovation as a web application powered by Next.js, strategically engineered to elevate both personal and professional efficiency. At its core, the tool seamlessly integrates smart task management features, enabling users to prioritize and organize their workflow intelligently. This innovation extends to automated calendar management, ensuring optimal scheduling for heightened time management. Collaborative project tools further enhance teamwork and coordination, fostering a dynamic environment. User-centricity is a key focus, with personalized insights into productivity trends, coupled with actionable recommendations to improve work habits. Embracing the ethos of continuous improvement, the Productivity Tool sets out to redefine how individuals and teams approach daily tasks, creating a pathway for innovation and efficiency.

2.2 Project Description: Brain Tumor Classification

This project focuses on creating an automated system for classifying brain tumors using advanced machine learning and deep learning methods. Brain tumors, which can be benign or malignant, require accurate and timely diagnosis to guide effective treatment plans. The project involves processing medical imaging data, primarily MRI scans, to develop models capable of identifying and categorizing tumors with high precision.

The workflow includes data collection, preprocessing (e.g., noise reduction and segmentation), feature extraction, and model training using algorithms like Convolutional Neural Networks (CNN). Evaluation metrics such as accuracy and precision are used to assess model performance. The end product is a user-friendly application designed to assist healthcare professionals in diagnosing brain tumors efficiently, reducing reliance on manual interpretation and enhancing diagnostic consistency. This innovation aims to improve patient outcomes and advance medical imaging applications.

The system leverages cutting-edge techniques, including deep learning architectures, to handle the complexities of brain imaging data. By employing data augmentation and transfer learning, the project addresses challenges such as limited dataset availability and variability in imaging quality. Additionally, the system incorporates explainable AI features to provide insights into its decision-making process, ensuring trust and transparency for medical professionals.

2. Existing/Legacy System

A legacy system for brain tumor classification typically uses traditional machine learning methods. Here's a simple overview:

1. Data Collection:

- MRI images of the brain are collected, often from public datasets.
- **Image Types:** MRI scans often consist of different types of sequences, such as T1-weighted, T2-weighted, and FLAIR, each providing different information about the tissue types in the brain.

2. Preprocessing:

- Images are resized, normalized, and enhanced to improve quality.
- **Resizing:** Images are resized to a consistent size (e.g., 256x256 pixels) so they can be processed uniformly by the machine learning model.

3. Feature Extraction:

- Important features like texture, shape, and intensity from the images are extracted using methods like edge detection and histograms.
- **Texture Features:** Methods like the **Gray Level Co-occurrence Matrix (GLCM)** are used to extract texture features, such as contrast, homogeneity, and entropy, which help distinguish tumor regions from surrounding tissue

4. Model Training:

- A machine learning model (like Support Vector Machine or k-NN) is trained using the extracted features to classify the tumor.
- The core model used is **Xception**, a pre-trained CNN available in the Keras library.

5. Classification:

- The model predicts if the tumor is benign, malignant, or of another type.
- **Prediction:** Once the model is trained, it is used to classify new MRI images based on the features extracted from those images. The model outputs a prediction, such as "benign," "malignant," or other predefined categories.

6. Evaluation:

- The model's accuracy is measured using metrics like accuracy, precision, and recall.
- The model uses **accuracy** as the primary evaluation metric during training and testing

2.4 Problem Statement

The current brain tumor classification system, based on deep learning and pre-trained models like Xception, faces several challenges that limit its effectiveness in real-world applications. These issues include:

1. **Data Limitations:**
    - Insufficient and imbalanced datasets lead to overfitting and biased predictions.
    - Variability in MRI imaging protocols reduces the model's ability to generalize across different sources.
2. **Model Limitations:**
    - High computational requirements restrict the system's use in low-resource environments.
    - Lack of interpretability makes it difficult for medical professionals to trust the predictions.
3. **Deployment Challenges:**
    - Real-time performance issues hinder deployment in critical healthcare scenarios.
    - Integrating the system into hospital workflows is complex due to compatibility and regulatory concerns.
4. **Clinical Impact:**
    - Misclassification can lead to incorrect treatments, causing severe consequences for patients.
    - The system struggles to generalize to unseen data, limiting its reliability in clinical use.

2.5 Need for a New System

To overcome these challenges, a new system is needed that addresses the following:

1. **Enhanced Data Handling:**
    - Access to larger and more diverse datasets.
    - Techniques to balance classes and reduce variability in input data.
2. **Improved Model Design:**
    - Lightweight and efficient models for real-time applications.
    - Explainable AI methods to provide insights into model predictions.

3. **Scalability and Integration:**
    - Models that can easily integrate with existing healthcare systems.
    - Scalable solutions for deployment across hospitals with varying resources.
4. **Clinical Validation:**
    - Rigorous testing and validation in real-world settings.
    - Collaboration with medical professionals to ensure the system meets clinical needs.

2.6 Problem Statement

The proposed brain tumor classification system addresses the limitations of existing methods by leveraging advanced data processing, optimized model design, and scalable deployment strategies. It incorporates diverse and augmented MRI datasets to improve generalization and reduce biases. A lightweight yet powerful deep learning model with explainability tools like Grad-CAM ensures accurate and interpretable predictions. Advanced techniques like transfer learning, class balancing, and robust evaluation metrics enhance reliability and performance. The system is designed for real-time use, compatible with both edge devices and cloud platforms. It also emphasizes data security and regulatory compliance, ensuring safe and ethical deployment. This solution aims to provide clinicians with a powerful tool to improve diagnostic accuracy and patient outcomes.

Key Components

1. **Data Preprocessing:**
    - Incorporates diverse and augmented MRI datasets with standardized preprocessing for consistent input quality.
2. **Model Design:**
    - Utilizes a lightweight, explainable deep learning architecture with attention mechanisms and transfer learning for accuracy and interpretability.
3. **Evaluation Metrics:**
    - Employs robust metrics like precision, recall, and ROC-AUC, along with cross-validation, to ensure reliable performance.
4. **Deployment Options:**
    - Supports real-time use through edge-compatible lightweight models and scalable cloud-based solutions.
5. **Data Security and Compliance:**
    - Ensures encryption and adherence to healthcare standards like HIPAA and GDPR for safe data handling.
6. **User Interface:**
    - Provides an intuitive platform for clinicians to easily upload MRI scans, view predictions, and access visual explanations.

2.7 Scope

The scope of brain tumor classification is vast and has significant implications for both medical practice and research. By leveraging advanced machine learning and AI techniques, brain tumor classification systems can transform how tumors are detected, diagnosed, and treated. These systems offer the potential for early diagnosis, improved treatment plans, and enhanced decision support for clinicians. They can be integrated into clinical workflows, making them accessible in hospitals and clinics worldwide. The system can also provide valuable insights for medical research, supporting the development of new therapies and improving patient outcomes.

Functional Scops:

- **Image Preprocessing:**
    - **Objective:** Ensure the quality of MRI images by resizing, normalizing, and enhancing images to standardize input data for the classification model.
    - **Functionality:** Apply preprocessing techniques such as noise reduction, resizing, and intensity normalization.

- **Tumor Detection and Classification:**
    - **Objective:** Automatically detect and classify the presence and type of brain tumors from MRI images.
    - **Functionality:** Use machine learning algorithms or deep learning models (e.g., CNNs) to classify tumor types such as glioma, meningioma, pituitary tumors, or no tumor.

- **Feature Extraction:**
    - **Objective:** Extract important features from the images that can help differentiate between different types of tumors.
    - **Functionality:** Employ techniques such as edge detection, texture analysis, and shape recognition to derive meaningful features.

User Scope:

• Patients: Patients can use the prediction model to predict the tumor.

• Doctors: doctors will be having data of patients and user management .

2.8 Outcomes of the Brain Tumor Classification System

The implementation of a brain tumor classification system delivers numerous significant outcomes, creating a positive impact across healthcare services, patient care, and medical research. By automating tumor detection and classification, the system addresses critical challenges in neuro-oncology, enabling improved efficiency, accessibility, and accuracy. These outcomes not only enhance clinical workflows but also empower healthcare providers with reliable decision-making tools and open new avenues for research and development in medical imaging.

Key Outcomes

1. **Improved Diagnostic Accuracy:**
    - Accurate classification of brain tumors significantly reduces the chances of misdiagnoses, leading to more precise and effective treatment plans.
    - The system ensures consistent and reliable results, even in cases where human interpretation may vary.
2. **Faster Diagnosis:**
    - Automated processes drastically cut down the time needed to analyze MRI scans, ensuring quicker response times for critical and emergency cases.
    - Faster diagnosis allows earlier treatment initiation, improving patient outcomes and reducing the burden of advanced-stage interventions.
3. **Personalized Treatment Plans:**
    - Detailed tumor classification enables clinicians to tailor treatment strategies based on the specific tumor type and stage.
    - This personalized approach improves treatment efficacy and reduces unnecessary procedures or generalized treatments.
4. **Enhanced Clinical Workflow:**
    - Seamless integration with existing hospital systems streamlines workflows, eliminating redundant manual processes and minimizing administrative overhead.
    - By automating tumor analysis, the system frees up medical professionals to concentrate more on direct patient care and strategic decision-making.
5. **Increased Accessibility:**
    - The scalable nature of the system ensures deployment in various settings, including resource-limited areas where access to expert radiologists may be restricted.
    - This accessibility reduces disparities in healthcare quality between urban and remote regions, benefiting underserved populations.

2.9 Tools & Technology with short justification

The mentioned tech stack comprises Flask , Python , Mongodb . Javascript , jupyter Notbook .

1. Flask :

Justification : Flask is a lightweight, open-source web framework for Python, designed to build web applications quickly and with flexibility.

2. Python :

Justification : Python is a versatile, high-level programming language known for its simplicity and readability. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming

3.MongoDB :

Justification : MongoDB is a NoSQL, document-oriented database that stores data in flexible, JSON-like documents. It is designed for scalability and high performance, making it ideal for handling large volumes of unstructured or semi-structured data.

4.Javascript :

JavaScript is a versatile, high-level programming language primarily used for creating interactive and dynamic web content. It runs in web browsers, enabling features like animations, form validation, and real-time updates

5.Jupyter Notbook :

Jupyter Notebook is an open-source, web-based interactive development environment that allows users to create and share documents containing live code, equations, visualizations, and narrative text.

Project Plan: Brain Tumor Classification System

1. Project Overview :

The Brain Tumor Classification System project aims to develop an intelligent and automated solution for accurately identifying and classifying brain tumors from MRI images. Leveraging advanced machine learning techniques, particularly deep learning, the system is designed to assist medical professionals in diagnosing tumors with high precision and speed. By integrating cutting-edge technologies, the project addresses critical challenges in neuro-oncology, such as diagnostic delays, misclassification, and limited accessibility to specialized healthcare services.

Key Deliverables :

1. User Authentication System

2. Responsive User Interface

3. User Management System

4. Deployment to hosrting plateforms

5. comprehensive documentation

6. Medical Diagnosis Detection

7. Accuracy of diagnosis

Project Timeline:

1**. Initiation Phase:**

• Project Definition Research: Clearly define the scope, objectives, and deliverables of the Notion Clone system, emphasizing productivity, collaboration, and user-friendly features.

• Stakeholder Identification: Identify and engage key stakeholders, including potential users, administrators, and technical teams, for requirements gathering and feedback.

2**. Planning Phase:**

• Requirement Analysis: Gather detailed requirements through surveys, interviews, and feedback sessions with potential users to understand their needs and expectations.

• System Design: Plan the system architecture, functionalities, and user interfaces, focusing on replicating and enhancing the features of Notion.

• Resource Allocation: Allocate human resources, budgets, and technological resources required for development, testing, and deployment

3.Design Phase:

• Database Planning: Plan database tables according to the modules and functionalities, ensuring efficient data storage and retrieval.

• Entity Relationship Diagram: Create a diagram illustrating the relationships between distinct entities within the database.

4. Development Phase:

• Frontend Development: Design and develop a user-friendly interface, ensuring ease of use and accessibility across various devices.

• Backend Development: Build the backend infrastructure for data management, processing, and integration with external systems.

5. Testing Phase:

• Unit Testing: Test individual components (frontend and backend) for functionality, performance, and compatibility.

• Integration Testing: Validate integration between system components and external educational systems.

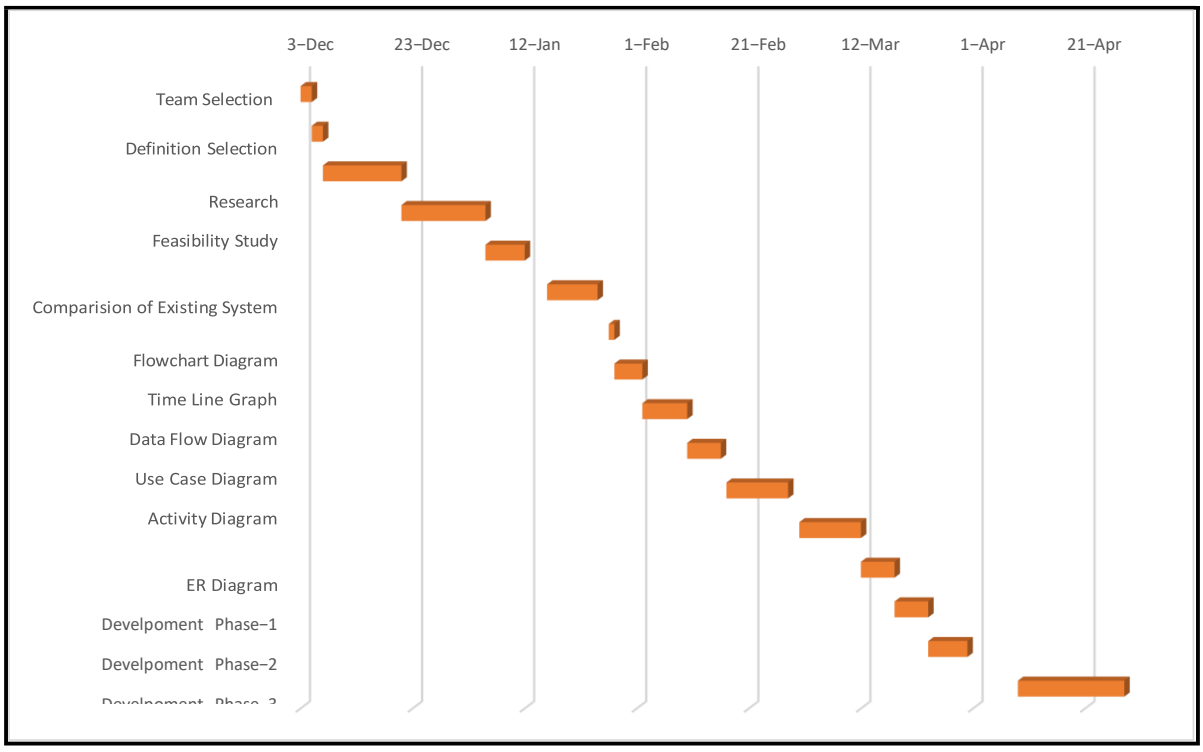
6. Deployment Phase:

• Deployment Strategy: Plan and execute the rollout strategy for the Notion Clone system implementation

• User Training: Conduct training sessions for users on system functionalities and best practices.

• System Go-Live: Deploy the Notion Clone system for live usage within the educational institution

Timeline Graph



Requirement Analysis

3.1 Feasibility Study

Feasibility is a study to evaluate feasibility of a proposed project or system. Feasibility study is one of the important four stages of the Software Project Management Process. Feasibility study is carried out based on many purposes to analyse whether a software product will be right in terms of development, implantation, contribution of project to the organization etc.

1. Technical Feasibility

- In Technical Feasibility current resources both hardware and software along with required technology are assessed to develop projects. This technical feasibility study reports whether there exists correct required resources and technologies which will be used for project development.

2. Economic Feasibility

- In the Economic Feasibility study the cost and benefit of the project is analysed. Means under this feasibility study a detailed analysis is carried out of what will be the cost of the project for development which includes all required cost for final development like hardware and software resource required, design and development cost and operational cost and so on.

3. Operational Feasibility

- In Operational Feasibility the degree of providing service to requirements is analysed along with how easy the product will be to operate and maintain after deployment. Along with this, other operational scopes are determining usability of the product, determining suggested solutions by the software development team is acceptable or not etc.

3.2 Users of the System

- **Patients: Patients will be able to get predicted health diseases with chances in the form of percentages , also they will be given suggetions and medicine advices**

- **Doctors : doctors will be having the full access of the patient's data also the predicted mri scans of the patients**

3.3 Modules of the System

1. **User Management Module :**

   Description : This module enables administrators to manage user roles, permissions, and access levels. Users can be categorized into roles of Patients

2. **Prediction Module :**

   Description : This Module will allow user to predict the tumor of the brain from MRI images , from that it will also give the suggetions by doctors

3. **Doctor suggestions :**

   Decsription : Doctors suggestion will be based on AI which analyze the tumor and from that It will automatically generate solutions for patients

4. **Knowledge Organization Module :**

   Description : Allows users to create databases, organize information, and establish relationships between different pieces of content

3.4 Process Model

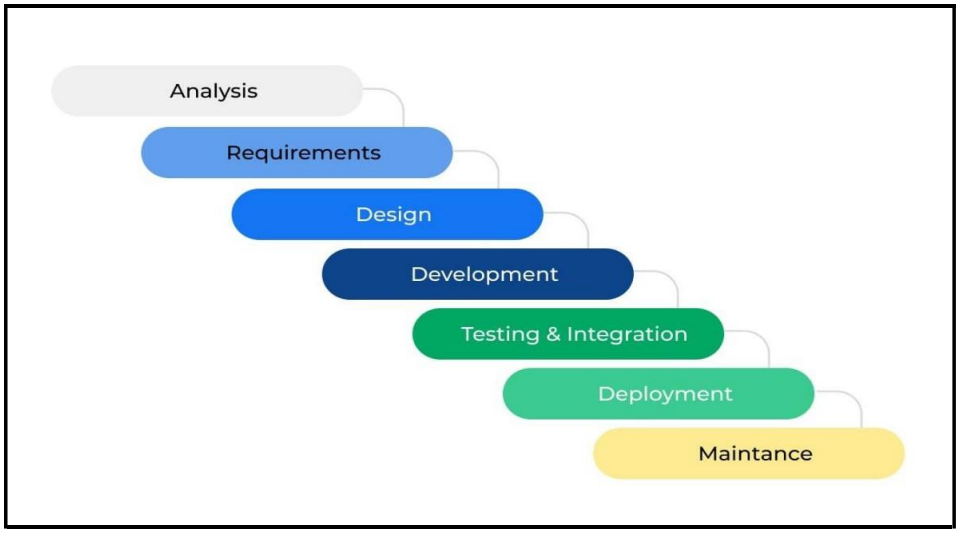


Figure a. [waterfall process model]

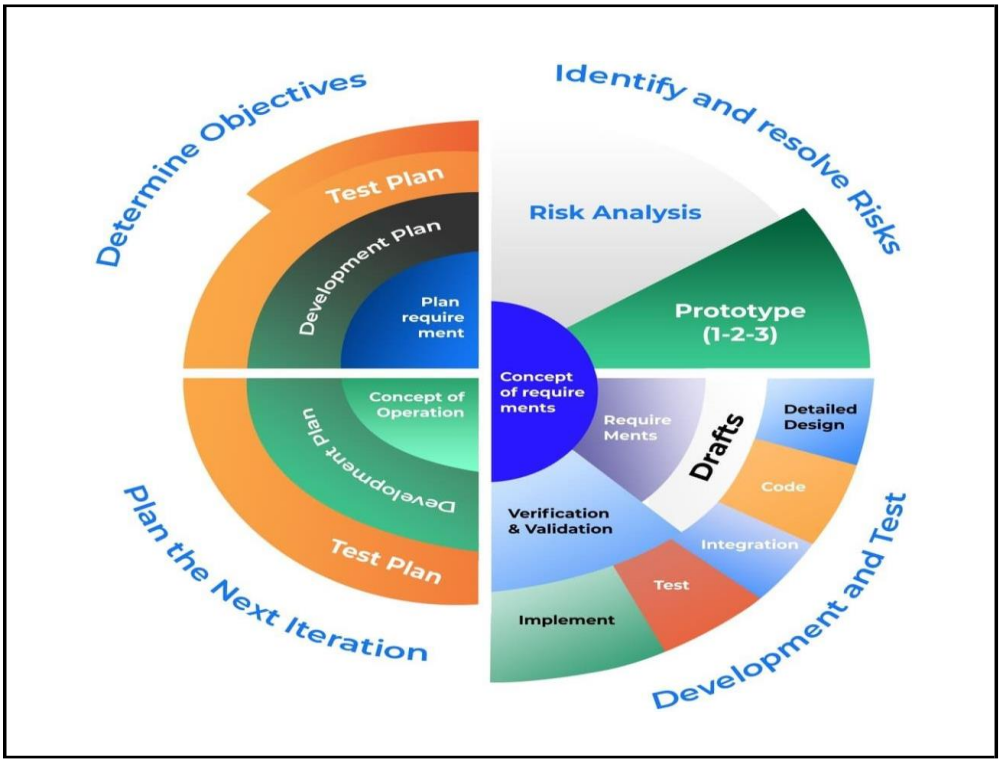


Figure b. [spiral process model]

3.5 Hardware & Software Requirements

Hardware Requirements (Admin/Dev)

Name of Components	Specification
Processor	Intel Core i5 / i7 or Higher
RAM	Minimum 4 GB or Higher
Storage	Minimum 512 GB SSD or Higher

Software Requirements (Admin/Dev)

Name of Components	Specification
Operating System	Windows 10 or Higher
Software development Kit	Next.js, React, Prisma ORM, many more…
Tools & Languages	Supabase, React Query, Shadcn UI, Tailwind CSS, TypeScript, JavaScript, many more…

Hardware Requirements (Others)

Name of Components	Specification
Processor	Intel Core i3 / i5 or Higher
RAM	Minimum 4 GB or Higher
Storage	Minimum 256 GB SSD or Higher

Software Requirements (Others)

Name of Components	Specification
Operating System	Windows 10 or Higher
Browser	Any Browser (Chrome, Brave, Arc, Safari etc)

4 Design

4.1 Use Case Diagram

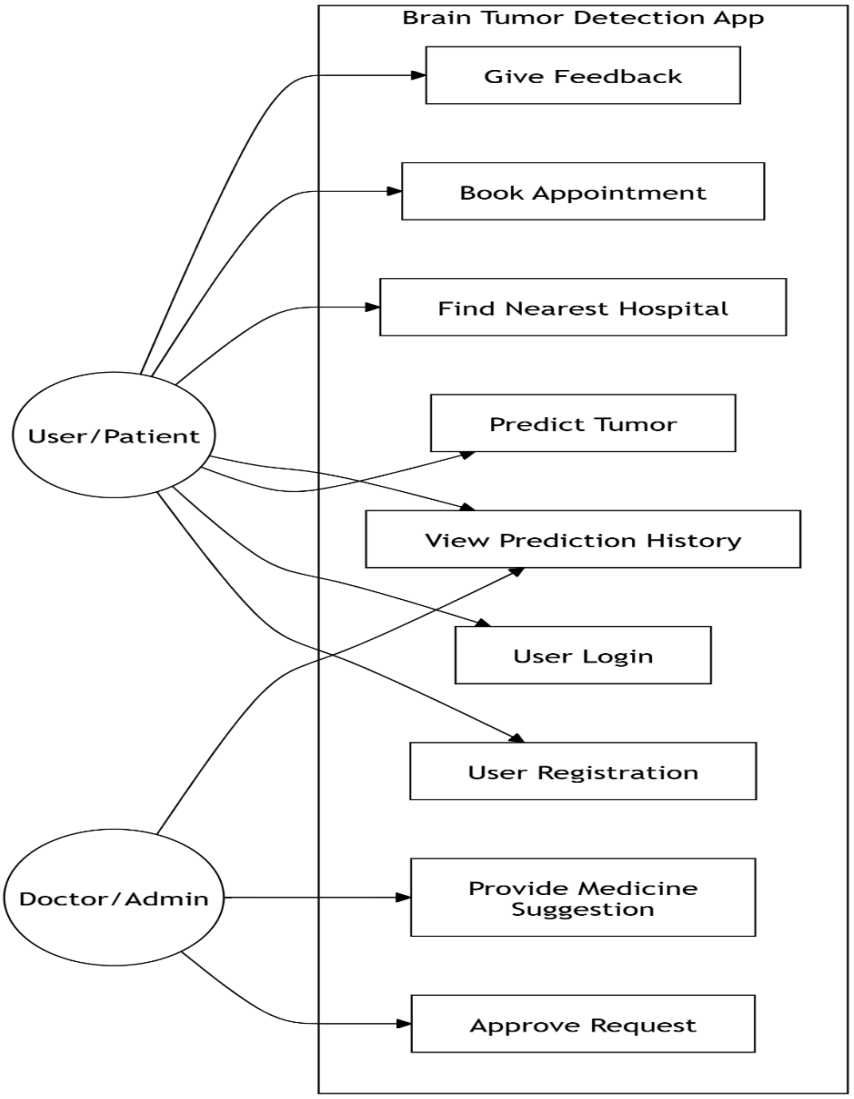


Figure [use case diagram]

4.2 Data Flow Diagram

Description of Data Flow Diagram

A Data Flow Diagram (DFD) is a graphical representation of how data moves within a system, illustrating processes, data stores, external entities, and the data flows connecting these components. At its core, a DFD is designed to showcase the flow of information throughout a system, emphasizing the transformations and interactions that occur.

DFDs are structured into different levels to provide a hierarchical view of the system. The levels include:

1. **Level 0 (Context Diagram):** This represents the entire system as a single process, surrounded by external entities. It offers a high-level overview of the system's boundaries and interactions with the external environment.

2. **Level 1:** This level provides a more detailed breakdown of the processes identified in the context diagram. Each process from the context diagram is decomposed into sub-processes, further refining the understanding of data flow within the system.

3. **Level 2, Level 3, and so on:** Subsequent levels continue the decomposition process, delving deeper into the details of processes until a satisfactory level of granularity is reached. These levels help in understanding the finer nuances of data flow and transformations within the system.

The hierarchical nature of DFDs allows for a step-by-step exploration of a system, making it an effective tool for understanding the flow and transformation of data in complex processes. DFDs are widely used in system analysis and design to document and communicate the functional aspects

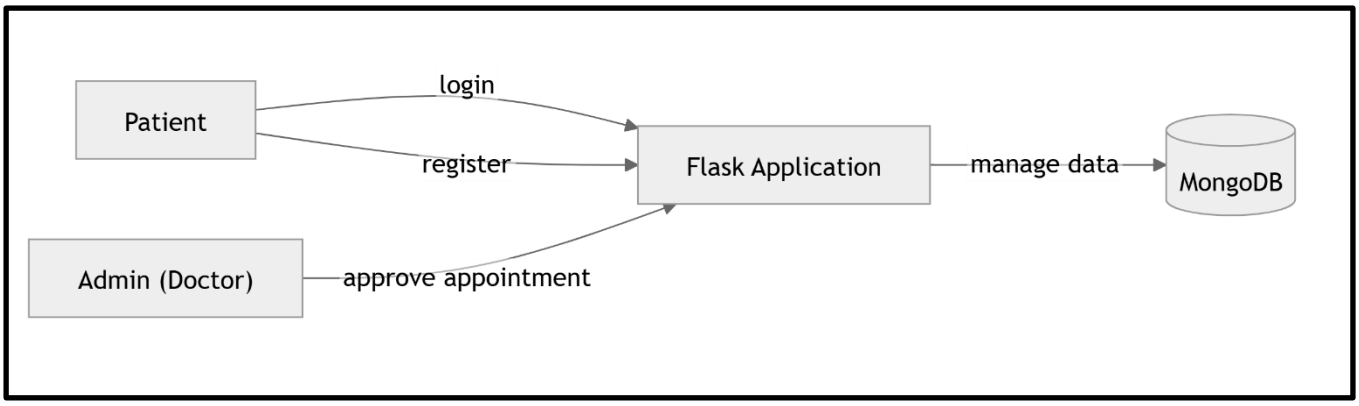


Figure a [level-0 data flow diagram]

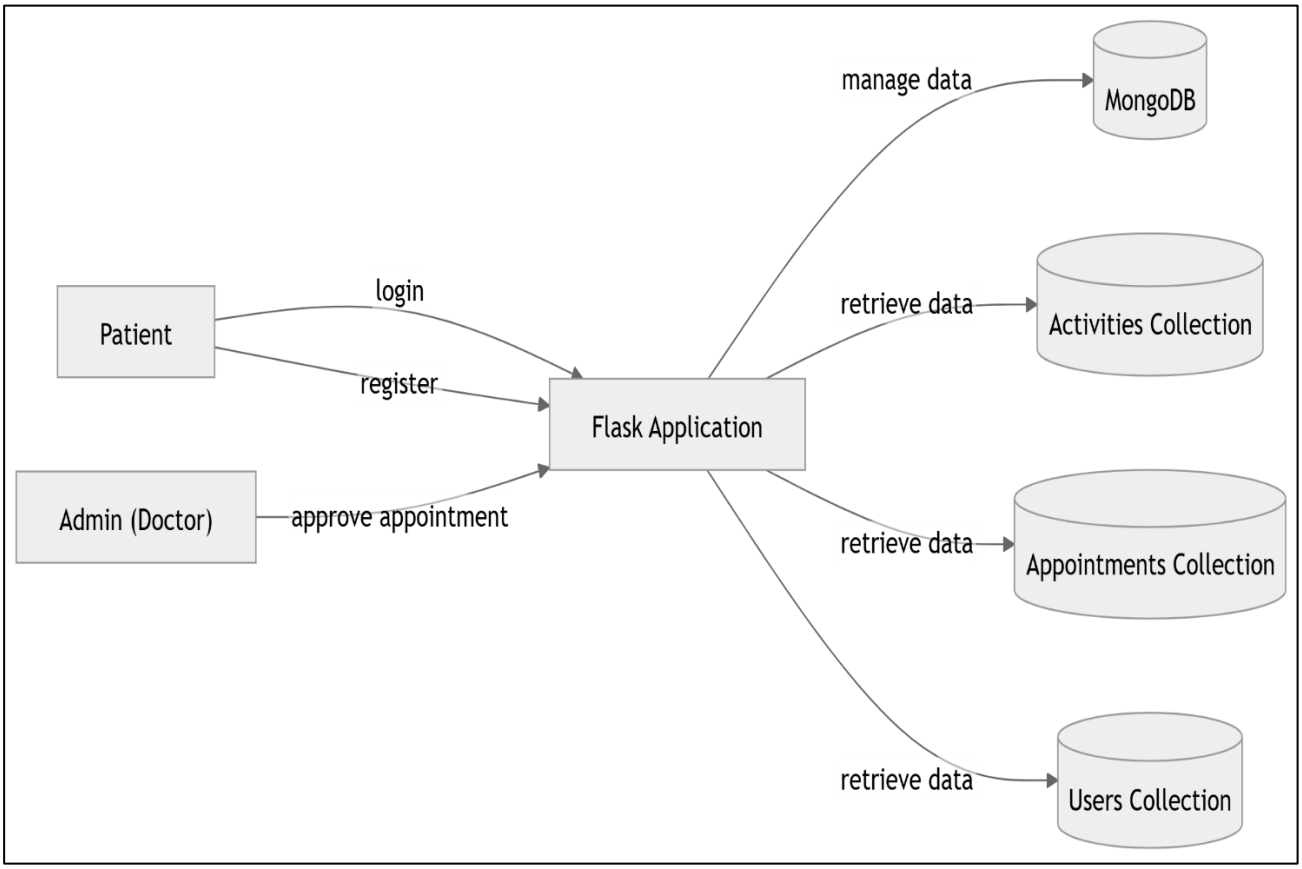


Figure b [level-1 data flow diagram]

4.3 Flowchart Diagram

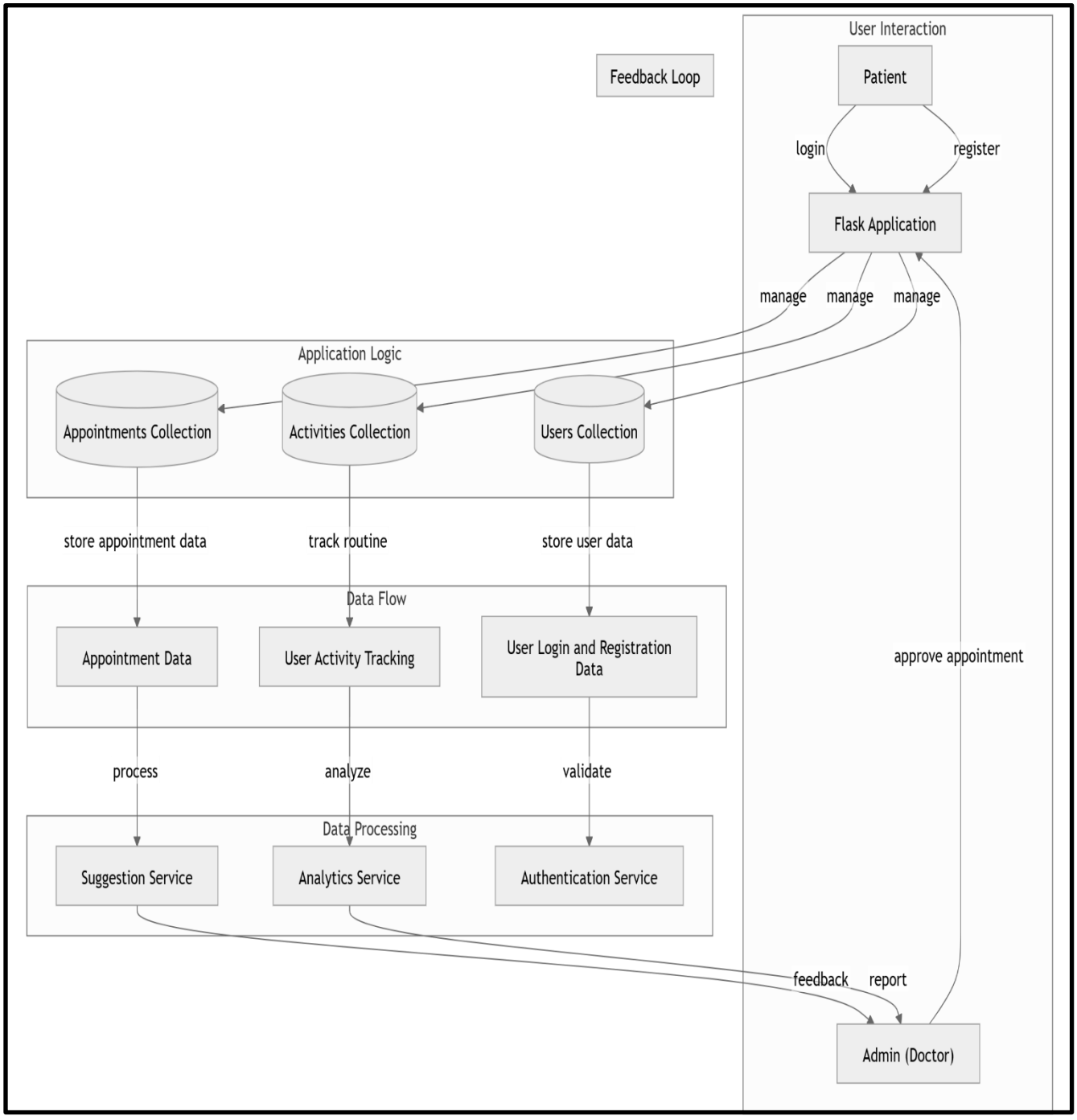


Figure c [level-2 data flow diagram]

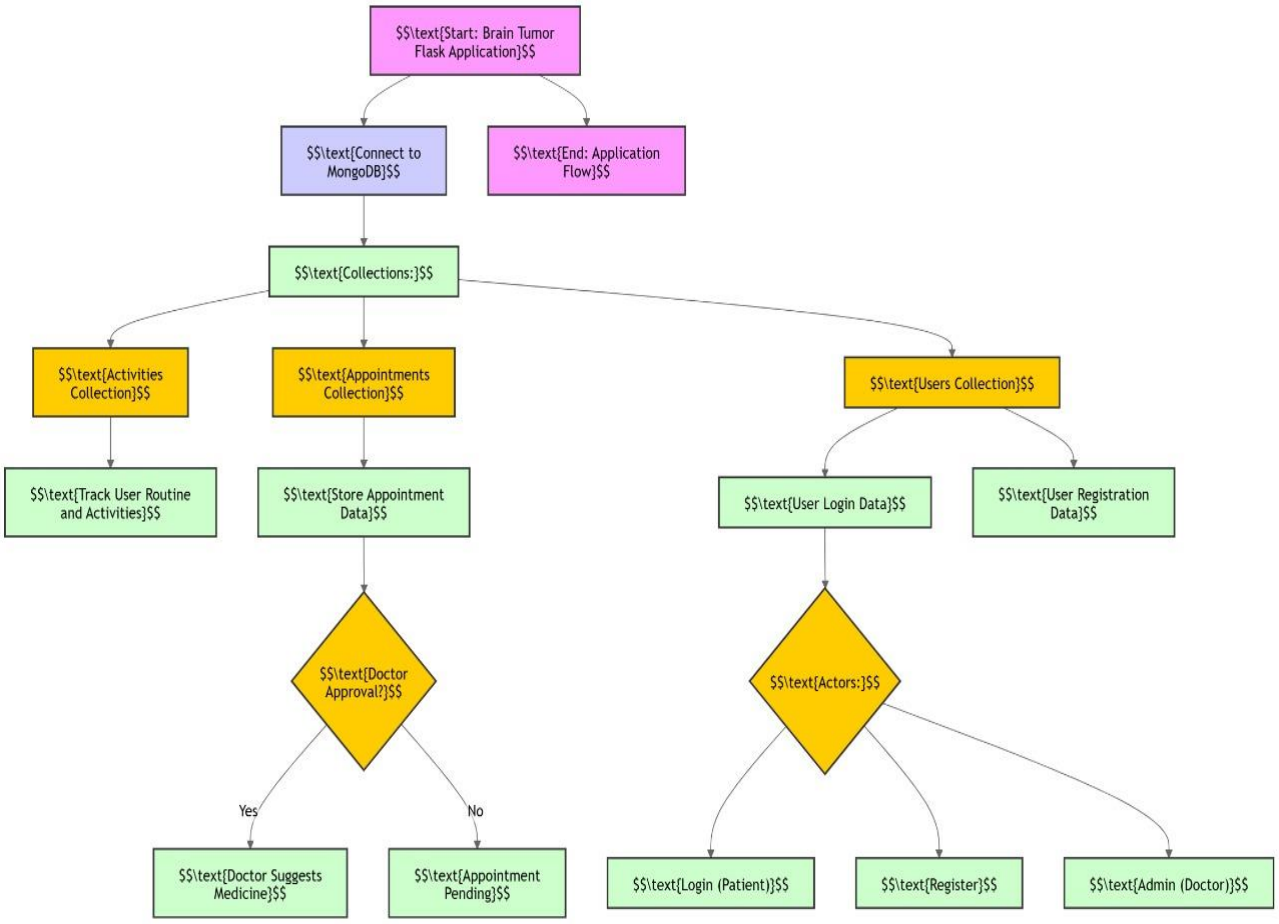


Figure [flow chart diagram]

Description of Flow Chart

A flowchart is a visual representation of a process or system, utilizing various shapes and arrows to illustrate the sequence of steps, decisions, and actions. Each shape in the chart represents a specific task or operation, and the arrows indicate the direction of flow between these elements. Flowcharts are widely used in various fields to diagram processes, analyse workflows, and facilitate understanding of complex systems through a simple and visually intuitive format. They serve as a valuable tool for planning, documentation, and communication in a concise and structured manner.

4.4 Class Diagram

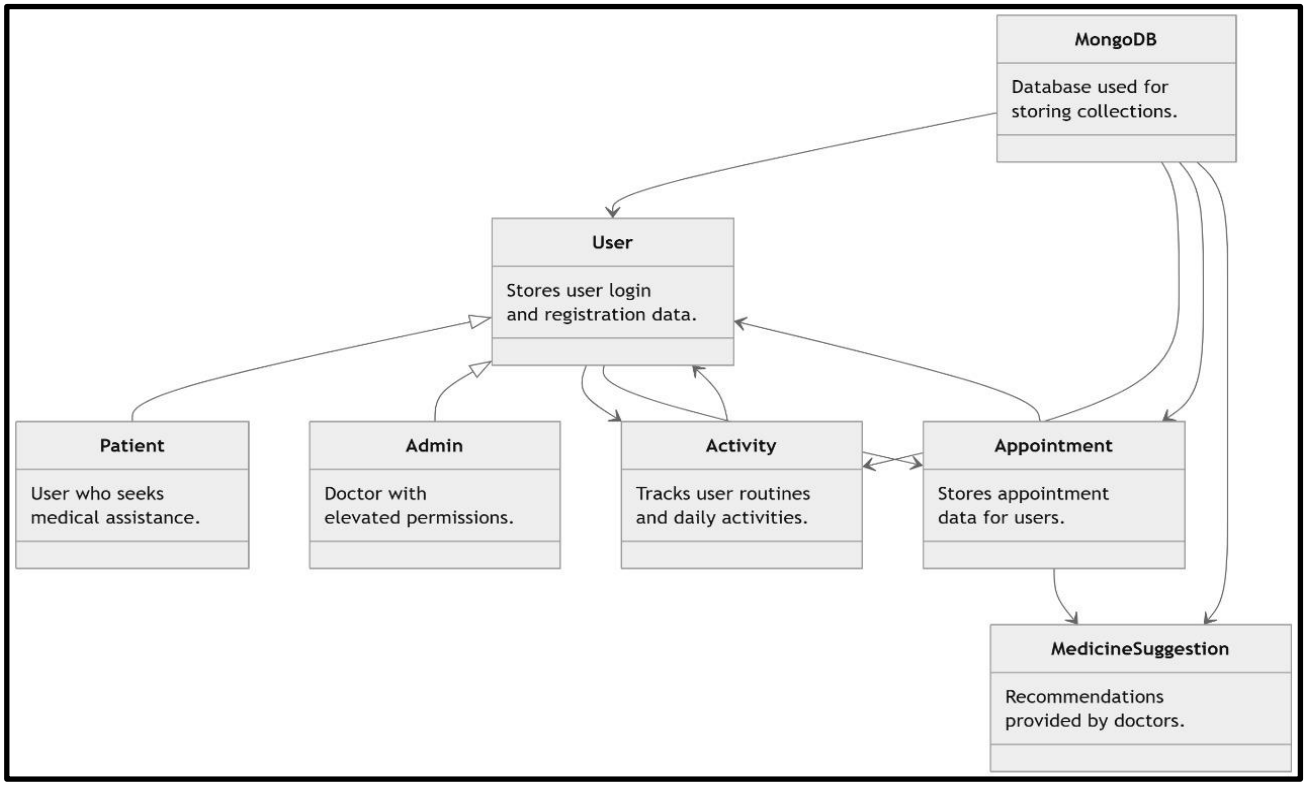


Figure a [class diagram]

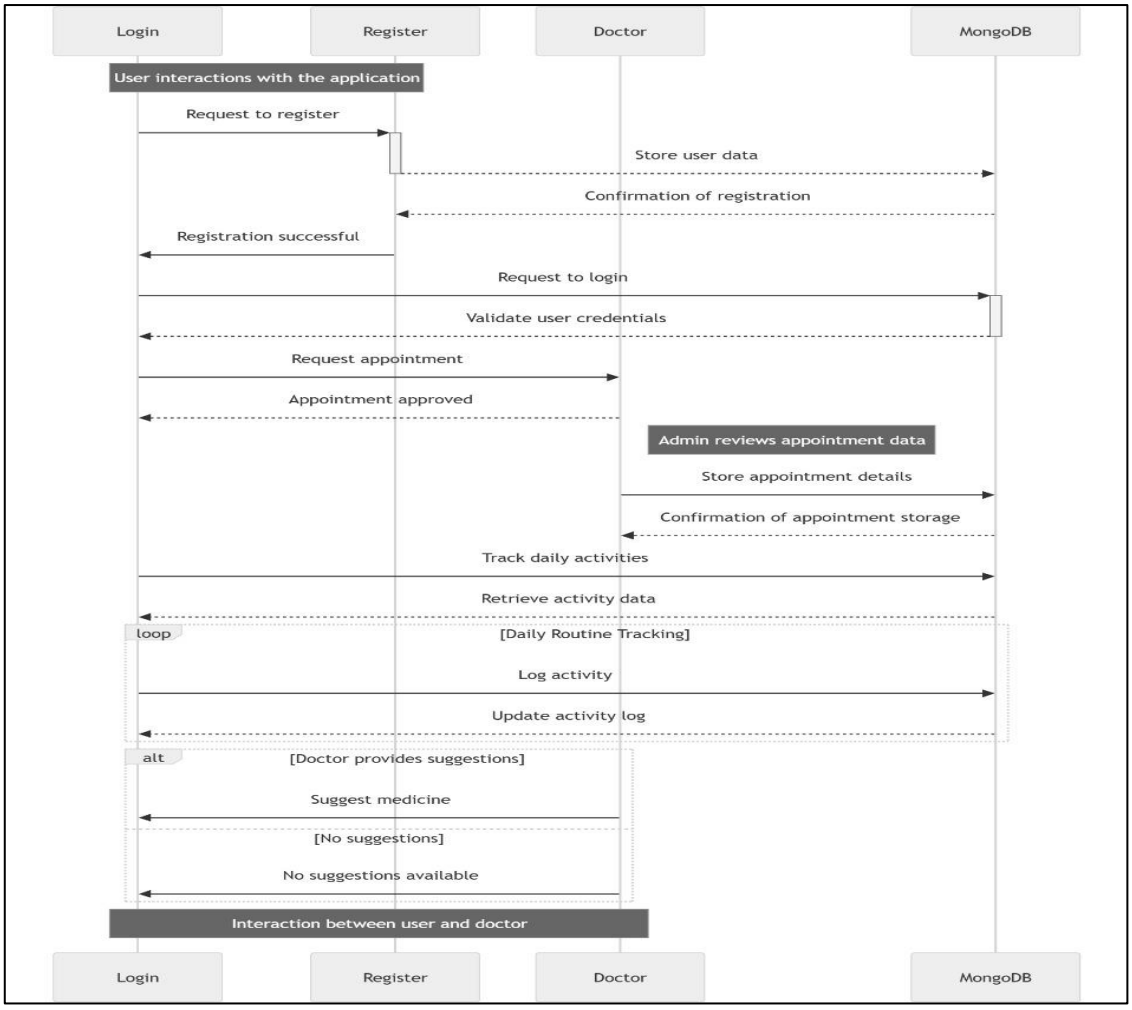
Description of Class Diagram

A class diagram is a visual representation in Unified Modelling Language (UML) that illustrates the structure of an object-oriented system. It showcases classes, their attributes, methods, and relationships, including associations, inheritance, and interfaces. Class diagrams provide a concise overview of the system's static structure, aiding in the design and communication of object-oriented software architecture.

A class diagram:

1. Illustrates the structure of an object-oriented system.
2. Represents classes, attributes, and methods.
3. Depicts relationships such as associations and inheritance.

4.5 Sequence Diagram



3 *Figure [sequence diagram]*

Description of Sequence Diagram

A sequence diagram visually shows how actors or objects interact through messages over time in a system. Lifelines represent participants, arrows depict message flow, and activation boxes show periods of activity. It's a concise tool for understanding dynamic system behaviour.

A sequence diagram:

1. Illustrates interactions and messages in a system.
2. Emphasizes the chronological order of actions.

3.1 Activity Diagram

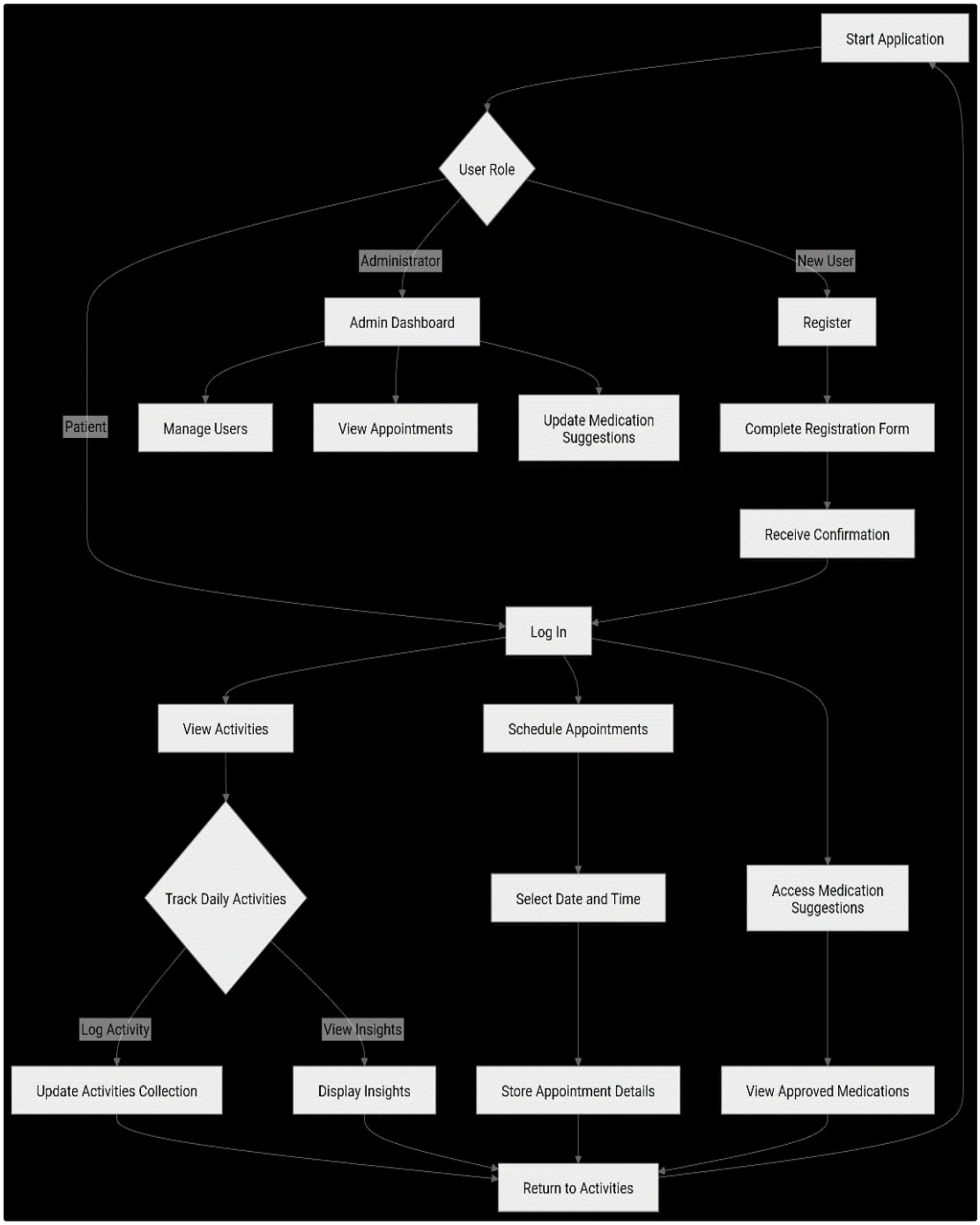
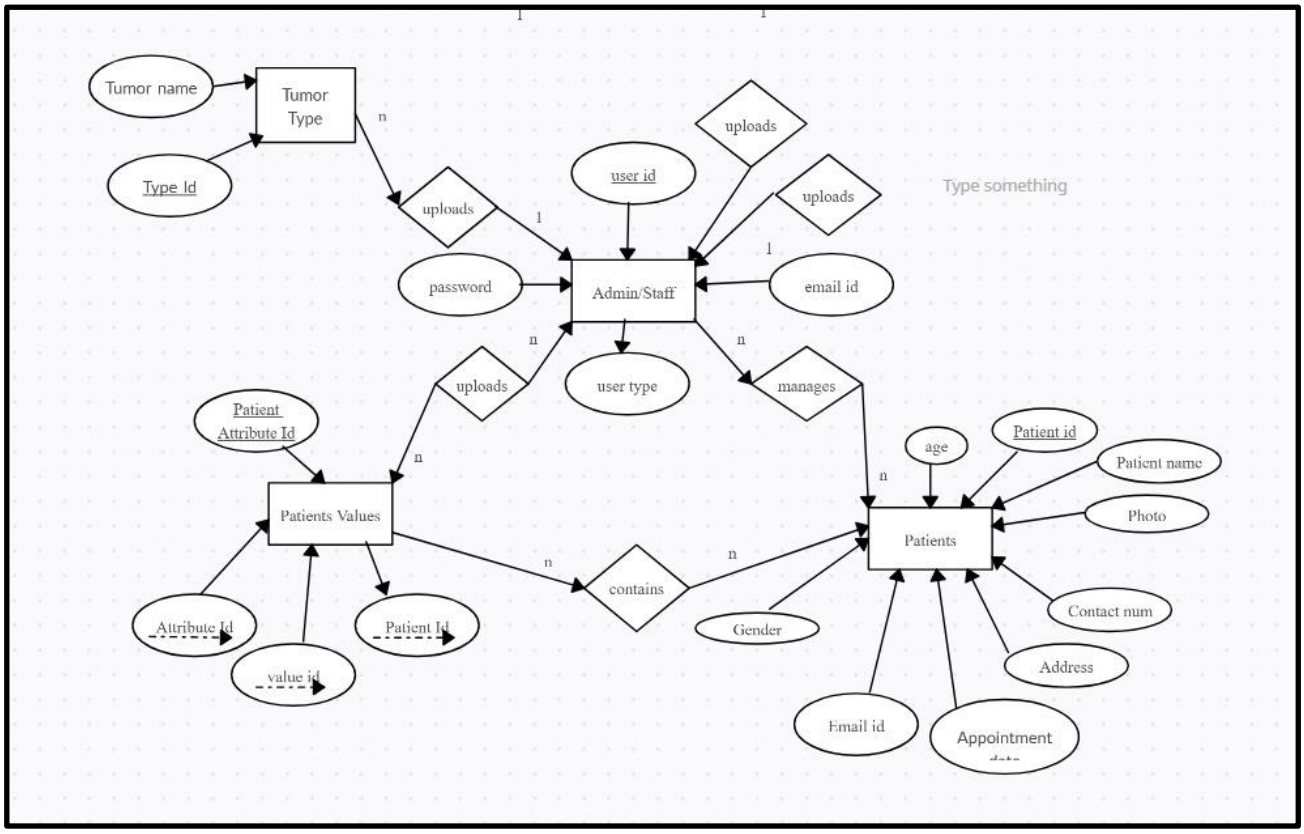


Figure [activity diagram

Entity Relationship Diagram



Description of Entity Relationship Diagram

An ER (Entity-Relationship) diagram is a visual representation that captures the logical structure of a database. It uses rectangles to represent entities (objects), diamonds for relationships, and lines to illustrate how entities are associated. Attributes, defining the properties of entities, are listed within the rectangles. The diagram aids in database design by showcasing the connections between different entities and their attributes, helping to understand the overall structure and relationships within the database model.

An Entity-Relationship (ER) diagram:

1. Illustrates the logical structure of a database.
2. Represents entities (objects) and their relationships.
3. Uses rectangles for entities, diamonds for relationships.

4.8 Data Dictionary

Description of Data Dictionary

A data dictionary is a centralized repository that provides concise and comprehensive information about the data elements used in a database or information system.

1. Table Name: users

Table Description: users table that stores the data by signup/login.

sr. no	field name	datatype	size	description	constraint	example
1.	id (primary)	int	11	unique identifier	primary key	1
2.	email	varchar	60	User's name	not null	abc@gmail.com
3.	password	varchar	60	Hashed password	not null	45j84ufiewfedsffdsfdf
4.	role	varchar	60	User role	not null	patient

2 .Table Name : activities

Table Description : it stores the user activities

sr. no	field name	datatype	size	description	constraint	example
1.	id (primary)	int	24	unique identifier	primary key	6795f5f956
2.	username	varchar	255	User's name	not null	test
3.	image_path	TEXT	-	Original image	not null	Any image
4.	marked_image	TEXT	-	Marked image	not null	Any image
5.	prediction	varchar	50	Tumor type predcition	not null	Glioma
6.	confidence	INT	-	Prediction confidence	Not null	100

3 . Table Name : appointments

Table Description : it stores the details of what user made like notes.

sr. no	field name	datatype	size	description	constraint	example
1.	id (primary)	int	24	unique identifier	primary key	6795f5f956e34137ced06ac7
2.	doctors_name	varchar	255	Doctor's name	not null	test
3.	date	DATE	-	Appointment date	not null	2025-02-01
4.	time	TIME	-	Appointment time	not null	12:00
5.	Patient_name	varchar	255	Patient's name	not null	test
6.	question	TEXT	-	Patient's query	Not null	"I have a tumor ."
7.	status	varchar	-	Appointment status	Not null	Approved
8.	medicine	varchar	50	Prescribed medicine	Not null	Paracetamol

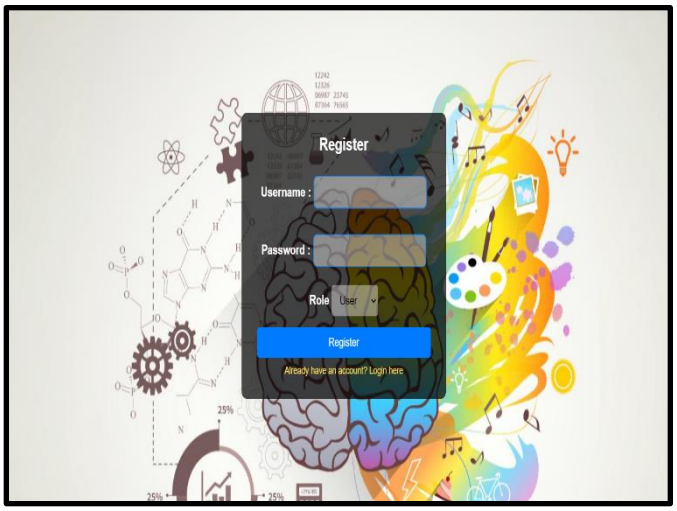
4 . Table Name : logs

Table Description : it stores the details of what user made like notes.

sr. no	field name	datatype	size	description	constraint	example
1.	id (primary)	int	24	unique identifier	primary key	6795f5f956e34137ced06a
2.	timestamp	DATETIME	-	Event timestamp	not null	2025-01-01 16:46:33
3.	action	varchar		User action type	not null	Login Attempt
4.	username	varchar	255	User's name	not null	test
5.	category	varchar	255	Patient's name	not null	patient
6.	status	varchar	50	User category	Not null	Success
7.	Additional_info	TEXT	-	Extra details	Not null	“ ”

5 Form Designing and Implementation

5.1 Form Layouts



Login and Register Form:

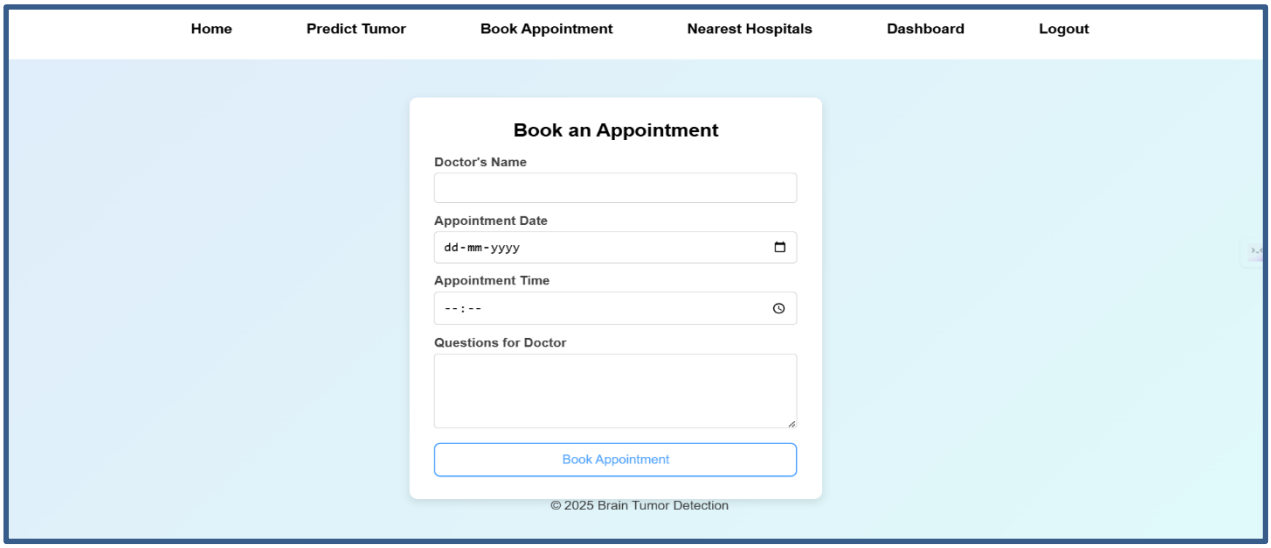
A login form, for existing users, verifies identity through a username/email and password combination, granting access to the account. Both forms are essential for user authentication and account management on websites or applications.

1. Register Form:

- Gathers user information for creating a new account.
- Collects essential details for user identification and communication.
- Often includes password creation and verification steps.
- May include additional fields depending on the platform's requirements (e.g., name, profile information).

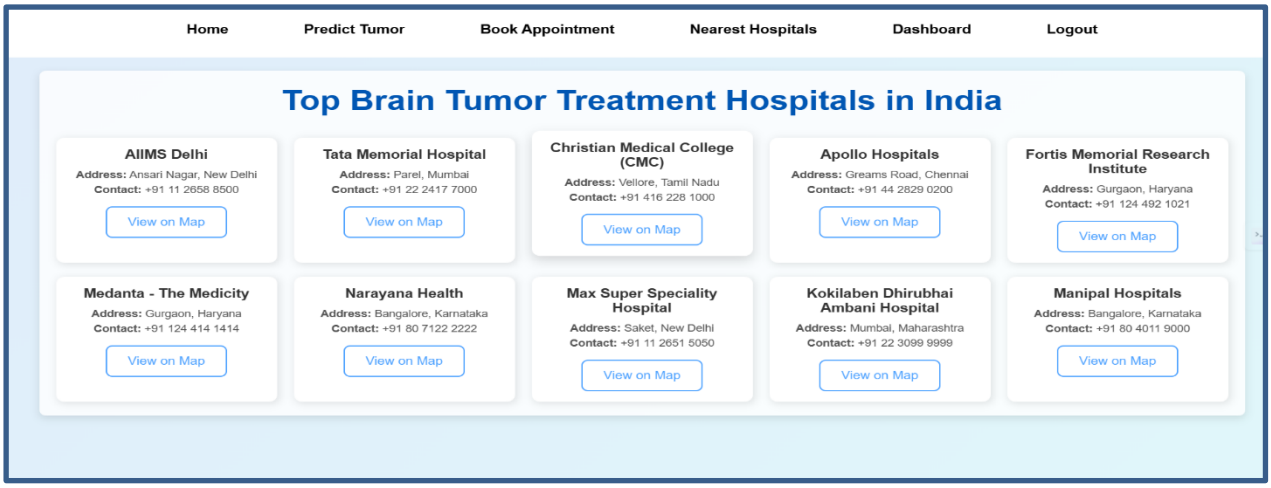
2. Login Form:

- Validates user identity through a combination of username/email and password.



Appointment Form :

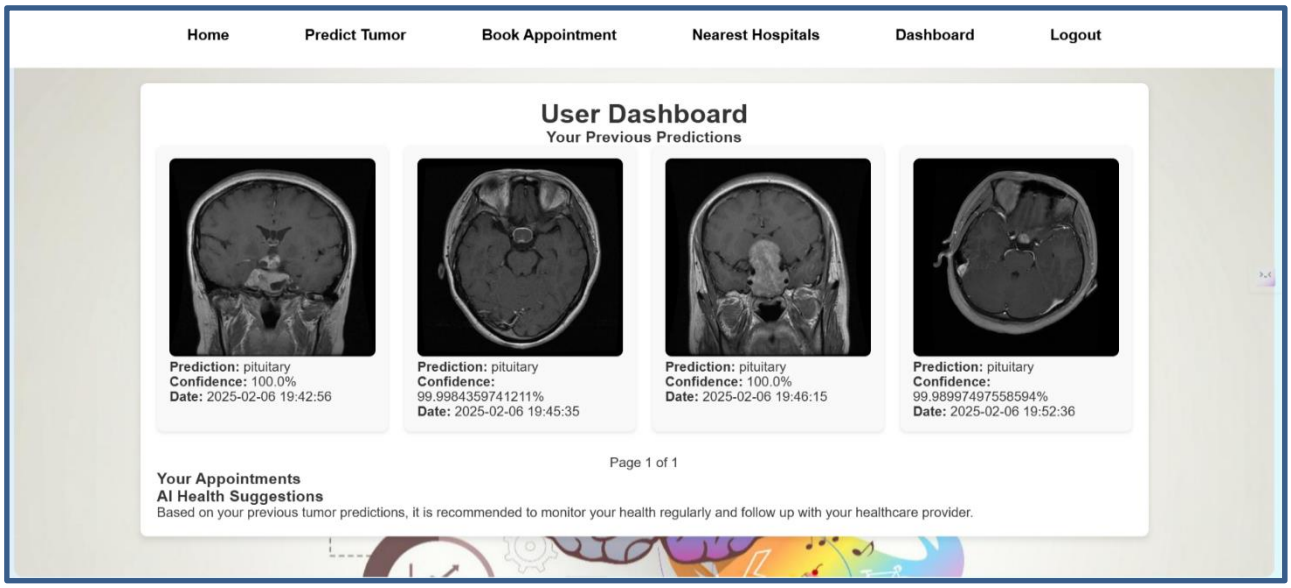
An appointment form is where users input and manage personal details such as Doctor's name , date and time of appointment including the section explaining the disease of brain .



Form to find nearest hospital :

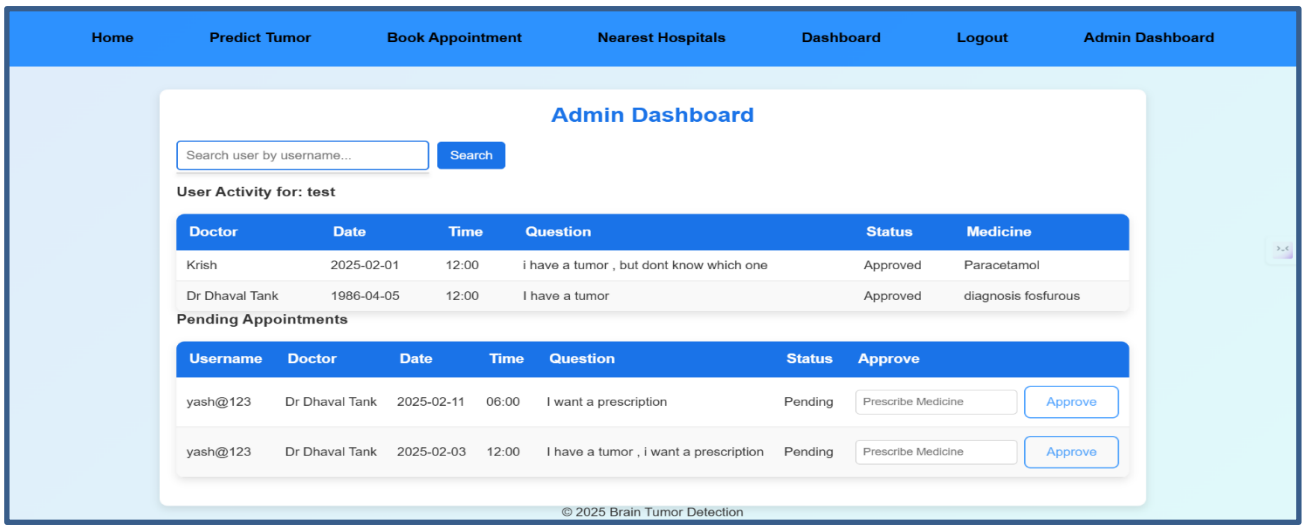
This form is where user can find their nearest hospital in google map and from the locations , user will be able to book the appointment for the treatment

5.2 Page Layout

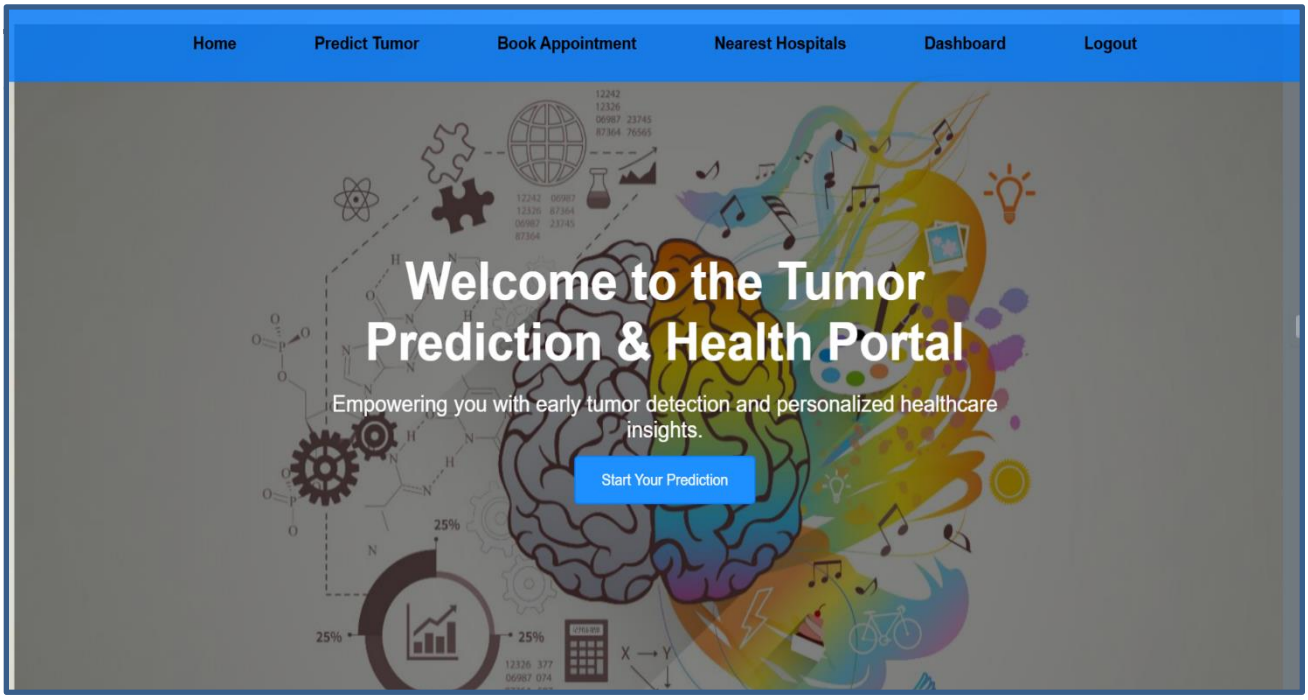


User Dashboard – Overview & Implementation

The User Dashboard is a key part of our web application that provides patients with a clear and structured interface to manage their medical history, tumor predictions, and appointments. It is designed to be user-friendly.



Admin Dashboard :



Dashboard Page:

A dashboard page serves as a central hub within a system or application, presenting users with a consolidated and visually intuitive overview of relevant information, metrics, or features. It often includes widgets, charts, and summaries that provide quick insights into key data points or functionalities. The dashboard is designed to enhance user experience by offering a comprehensive snapshot of critical aspects, such as analytics, performance metrics, or personalized content, allowing users to efficiently monitor and manage various aspects of the system from a single interface.

- User Authentication Links – Provides easy access to Login and Register buttons for new and existing users.
- Brain Tumor Prediction Service – Highlights the AI-powered tumor detection system, guiding users to upload MRI scans for diagnosis.
- Appointment Booking – Encourages users to book an appointment with a specialist for further consultation.
- Informative Content – Gives a brief introduction to the platform and its capabilities in brain tumor detection.

5.3 Coding Convention

```python
class LoginForm(FlaskForm):
    username = StringField('Username', validators=[InputRequired()])
    password = PasswordField('Password', validators=[InputRequired()])

@app.route('/login', methods=['GET', 'POST'])
def login():
    form = LoginForm()
    if form.validate_on_submit():
        username = form.username.data
        password = form.password.data
        user = mongo.db.users.find_one({"username": username})

        if user and check_password_hash(user['password'], password):
            login_user(User(user['_id'], user['username'], user.get('role', 'user')))

            # Redirect Admins to the Admin Dashboard
            if user.get('role') == 'admin':
                return redirect(url_for('admin_dashboard'))

            return redirect(url_for('dashboard'))

        flash('Invalid credentials!', 'danger')

    return render_template('login.html', form=form)


@app.route('/dashboard')
@login_required
def dashboard():
    # Fetch user activities (tumor prediction history) from the database
    activities = mongo.db.activities.find({"username": current_user.username})  # Assuming activities are linked by username

    # Fetch the user's appointments
    appointments = mongo.db.appointments.find({"patient_name": current_user.username})

    # Generate an AI suggestion for the user
    ai_suggestion = "Based on your previous tumor predictions, it is recommended to monitor your health regularly and follow up with your healthcare provider."

    # Paginate the results
    page = request.args.get('page', 1, type=int)
    per_page = 8
```

```python
    page = request.args.get('page', 1, type=int)
    per_page = 8
    total_activities = mongo.db.activities.count_documents({"username": current_user.username})  # Use count_documents instead of count
    total_pages = (total_activities + per_page - 1) // per_page
    activities_paginated = activities.skip((page - 1) * per_page).limit(per_page)

    return render_template('user_dashboard.html',
                           activities=activities_paginated,
                           appointments=appointments,
                           ai_suggestion=ai_suggestion,
                           page=page,
                           total_pages=total_pages)

# Fetch User Suggestions (AJAX)
@app.route('/search_user_suggestions', methods=['GET'])
@login_required
def search_user_suggestions():
    if current_user.role != 'admin':
        return jsonify([])

    search_term = request.args.get('query', '').strip()
    if not search_term:
        return jsonify([])

    users = mongo.db.users.find({"username": {"$regex": search_term, "$options": "i"}}, {"username": 1}).limit(5)
    user_list = [user["username"] for user in users]

    return jsonify(user_list)


# Admin Dashboard Route
@app.route('/admin_dashboard', methods=['GET', 'POST'])
@login_required
def admin_dashboard():
    if current_user.role != 'admin':
        flash('Unauthorized access!', 'danger')
        return redirect(url_for('index'))

    # Pagination settings
    page = int(request.args.get('page', 1))
    per_page = 5
    skip = (page - 1) * per_page
```

```python
@app.route('/nearest_hospitals')
@login_required
def nearest_hospitals():
    # Example: This is just a mock list of hospitals for demonstration
    hospitals = [
        {"name": "City Hospital", "address": "123 City St.", "contact": "123-456-7890"},
        {"name": "Central Medical Center", "address": "456 Central Ave.", "contact": "987-654-3210"},
        {"name": "HealthCare Clinic", "address": "789 Health Rd.", "contact": "555-123-4567"}
    ]

    return render_template('nearest_hospitals.html', hospitals=hospitals)


from datetime import datetime

@app.route('/predict', methods=['GET', 'POST'])
@login_required
def predict():
    if request.method == 'POST':
        file = request.files['image']
        if file and allowed_file(file.filename):
            filepath = os.path.join(app.config['UPLOAD_FOLDER'], secure_filename(file.filename))
            file.save(filepath)
            img = preprocess_image(filepath)
            prediction = model.predict(img)
            class_index = int(np.argmax(prediction))
            class_label = labels[class_index]
            confidence = float(prediction[0][class_index] * 100)

            # Save the prediction result to MongoDB
            mongo.db.activities.insert_one({
                "username": current_user.username,
                "prediction": class_label,
                "confidence": confidence,
                "date": datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
                "image": file.filename  # Save the image filename in the database
            })
            return render_template('result.html', label=class_label, confidence=confidence, filepath=filepath)
        flash("Invalid file type!", "danger")
    return render_template('predict.html')
```