Project Report On

"Predico – Disease Prediction System"

Name	Regd No	Branch
Aman Kumar Pattanaik	2301030051	CSE
Prasuna Dash	2301030048	CS-IT
B Raj Kumar Patro	2301030023	CS-IT
Subham Rout	2301030098	CS-IT
Aliva Priyadarshini Sahoo	2301030069	CS-IT
Baijayanti Mahanta	2301030125	COE

Contents

1. Introduction
2. Objective
3. Scope
4. Features
5. Technology Stack
6. System Architecture
7. Dataset and Data Collection
8. Machine Learning Model
9. System Interfaces
10. Challenges
11. Learnings
12. Future Scope
13. Conclusion
14. References

1.Introduction:

Predico is an innovative web-based platform that bridges the gap between early disease detection and timely medical consultation. Built using Django and powered by advanced machine learning algorithms, Predico empowers users to actively participate in their healthcare journey. By allowing individuals to input their symptoms and instantly receive potential diagnoses, the system plays a crucial role in early intervention and preventive care. In addition to disease prediction, the platform offers seamless access to online consultations with recommended doctors, thus ensuring that medical advice is just a click away. Predico is thoughtfully designed to cater to the unique needs of patients, healthcare professionals, and administrators, all while maintaining a user-friendly interface and secure data environment.

2. Objective:

- To harness the power of machine learning to accurately predict potential diseases based on diverse and dynamic symptom inputs from users.

- To enhance healthcare accessibility by offering real-time online consultations with qualified medical professionals.
- To implement a secure, role-based digital ecosystem that maintains data integrity and ensures appropriate user access across patients, doctors, and administrators.
- To reduce delays in diagnosis and encourage early intervention, ultimately contributing to improved patient outcomes and reduced burden on traditional healthcare systems.

3. Scope:

Predico has been carefully structured to address the healthcare interaction needs of multiple user categories:

- **Patients**: Users can effortlessly register and securely log in to access a guided interface where they can input their symptoms. Based on the inputs, the system utilizes its trained machine learning model to generate a probable disease prediction. Patients can then view a curated list of suggested doctors and initiate online consultations.

- **Doctors**: After logging in, medical professionals can access patient consultation requests, evaluate the inputted symptoms, and provide medical advice or follow-up actions. Doctors can also maintain records and review consultation histories for better continuity of care.
- **Administrators (Admins)**: Responsible for the platform's integrity, administrators have the authority to manage user accounts, approve doctor credentials, monitor consultation logs, and ensure overall system performance.

4. Features:

Predico offers a rich set of functionalities that enhance the healthcare experience for all user roles:

- **Advanced Disease Prediction**: Users can enter symptoms from a predefined list, and the system leverages a trained machine learning model to provide potential diagnoses.
- **Smart Doctor Suggestions**: Once a disease is predicted, relevant doctors are recommended to the patient, streamlining the consultation process.

- **Role-Based Access**: With separate access levels for patients, doctors, and admins, the system ensures that sensitive data is protected and accessible only by authorized users.
- **Seamless Online Consultation**: The platform supports digital interaction between patients and doctors through a dedicated interface, facilitating remote healthcare delivery.
- **Secure User Management and Logging**: The system tracks user activities and maintains history logs for future reference and accountability.

5. Technology Stack:

The development of Predico integrates a range of modern web technologies and data science tools to ensure a seamless and efficient user experience:

- **Frontend Technologies**: HTML, CSS, and Bootstrap form the layout and design framework, while JavaScript and jQuery manage dynamic content and user interactions.
- **Backend Framework**: Django, a robust Python-based framework, handles all server-side logic, including routing, authentication, and form handling.

- **Database**: PostgreSQL is used for storing structured data, including user profiles, consultation logs, and system configurations.
- **Supporting Tools**: PgAdmin is employed for database management, while Orange is used for initial machine learning experimentation and visualization.
- **Machine Learning Libraries**: scikit-learn is used for model training and testing, and joblib for model serialization and loading.

6. System Architecture:

Predico's system architecture is designed to ensure high reliability, modularity, and scalability:

- **Frontend Interface**: Separate user dashboards are implemented for patients, doctors, and administrators, each with specific functionalities relevant to the user's role.
- **Backend Services**: The Django framework handles core logic, data manipulation, authentication, and serves machine learning predictions via API endpoints.

- **Machine Learning Integration**: The ML model is trained separately and then integrated with Django using joblib for runtime predictions.
- **Database Layer**: PostgreSQL serves as the relational database for managing users, symptoms, disease predictions, and consultation records in a normalized structure.

7. Dataset & Data Collection:

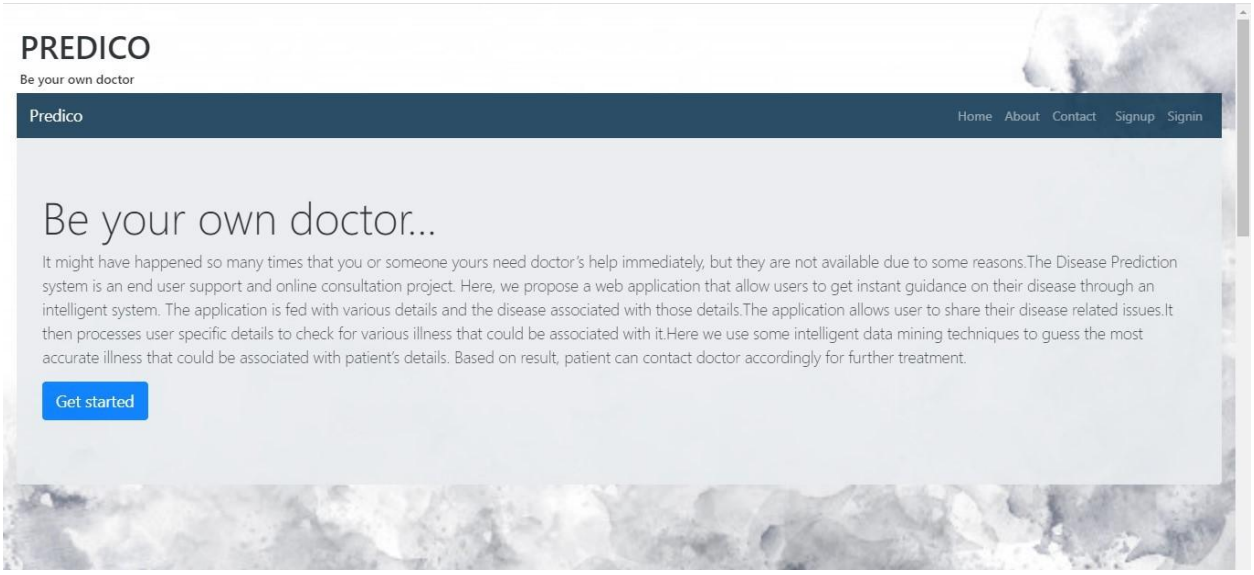
- Dataset collected from Kaggle and other verified medical websites.
- **Total Records**: 5000 patient entries.
- **Symptoms**: 132 types
- **Disease Classes**: 40 general diseases (e.g., Diabetes, Malaria, Jaundice, etc.)
- Data is real and medically validated, ensuring no dummy values.

8. Machine Learning Model:

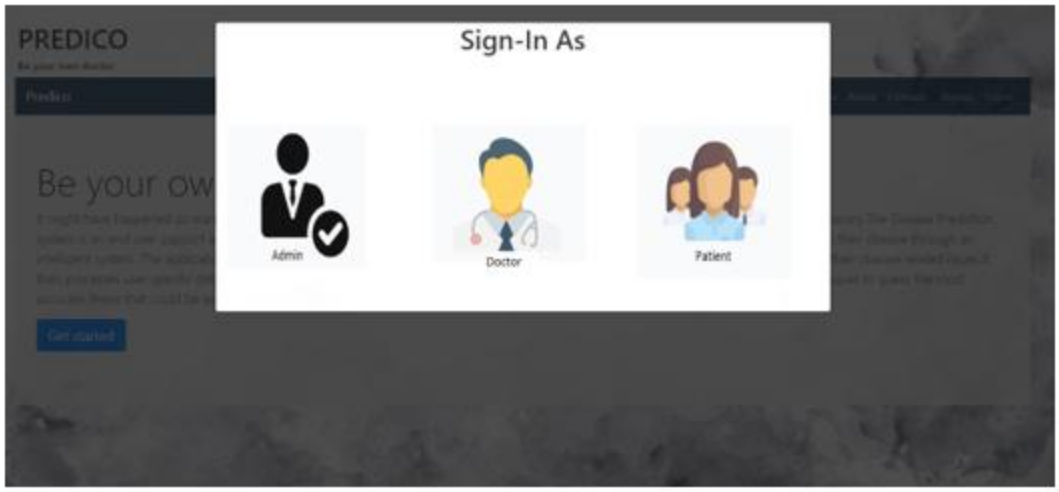
- Trained using **scikit-learn** models like Decision Tree, Random Forest.
- Model evaluated using cross-validation.

- Model serialization done using **joblib** for quick deployment.
- Integrated with Django to provide real-time predictions on symptom input.

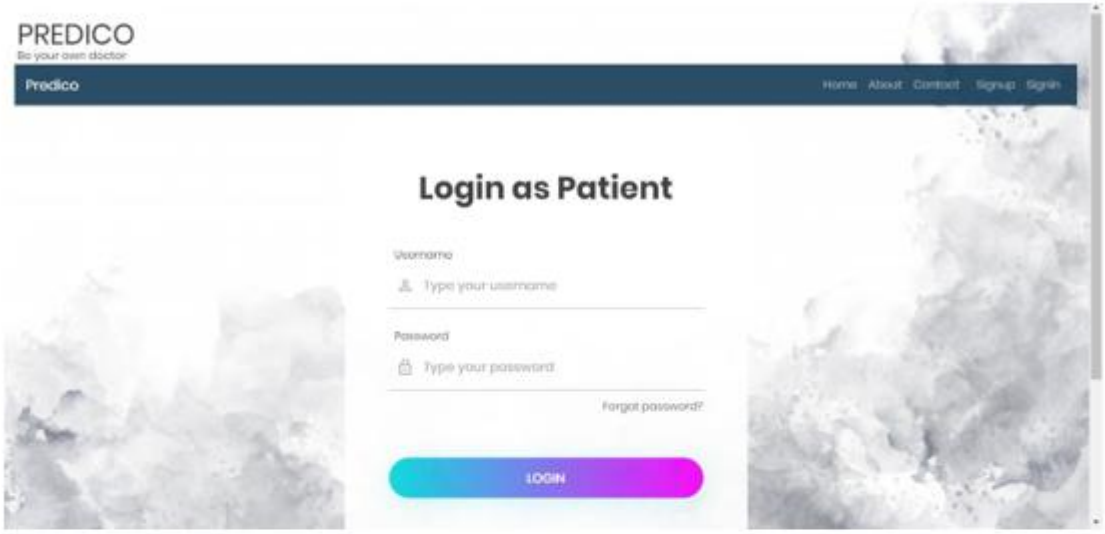
9. System Interfaces:



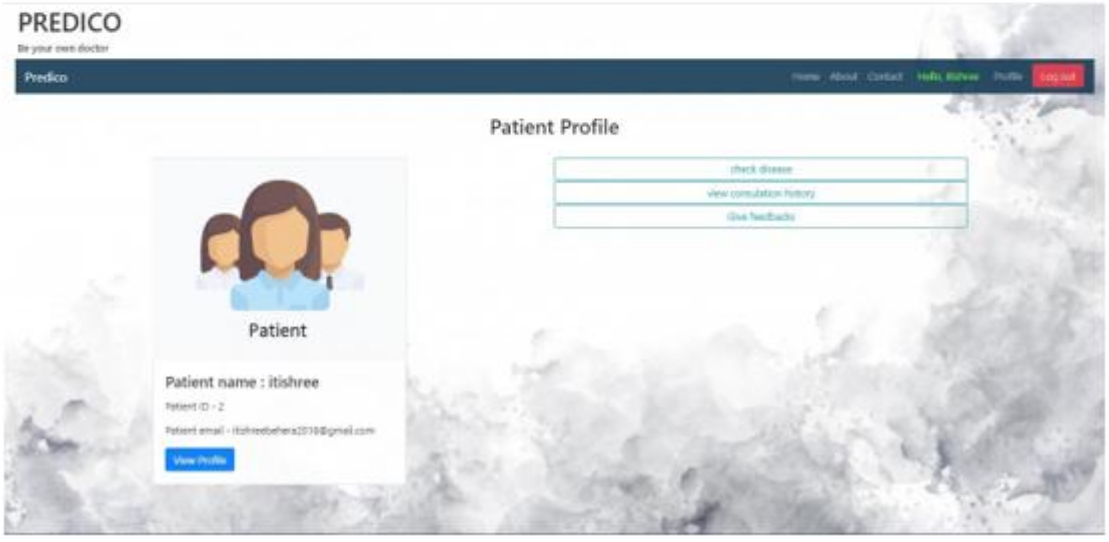
Login Modal-



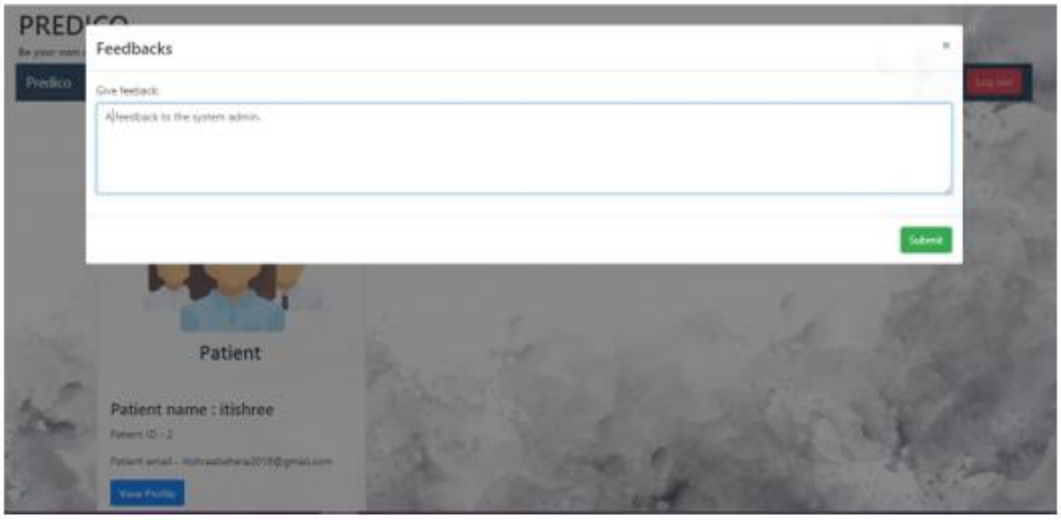
Login as Patient-



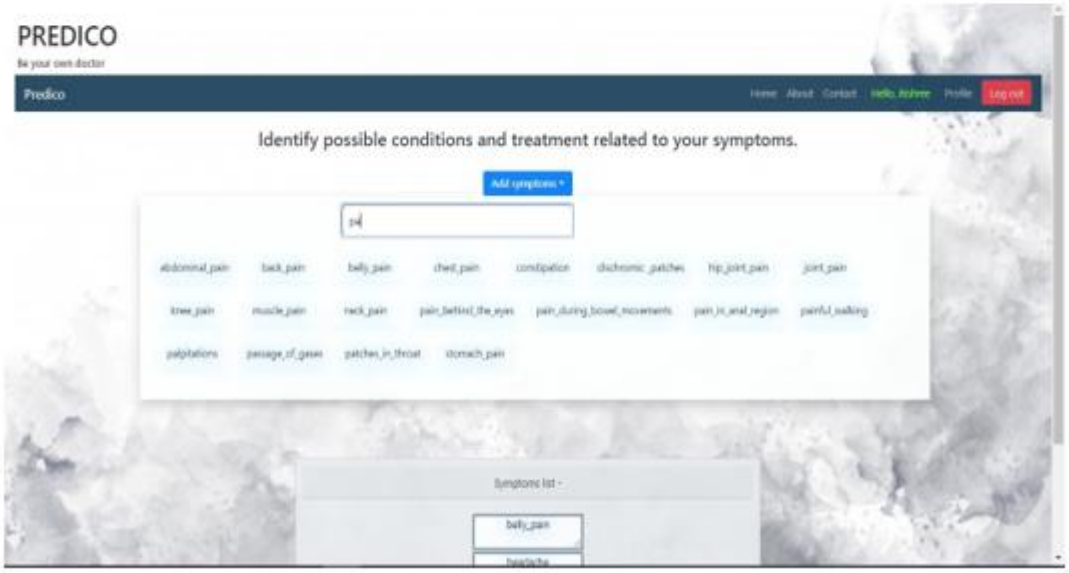
Patient UI-



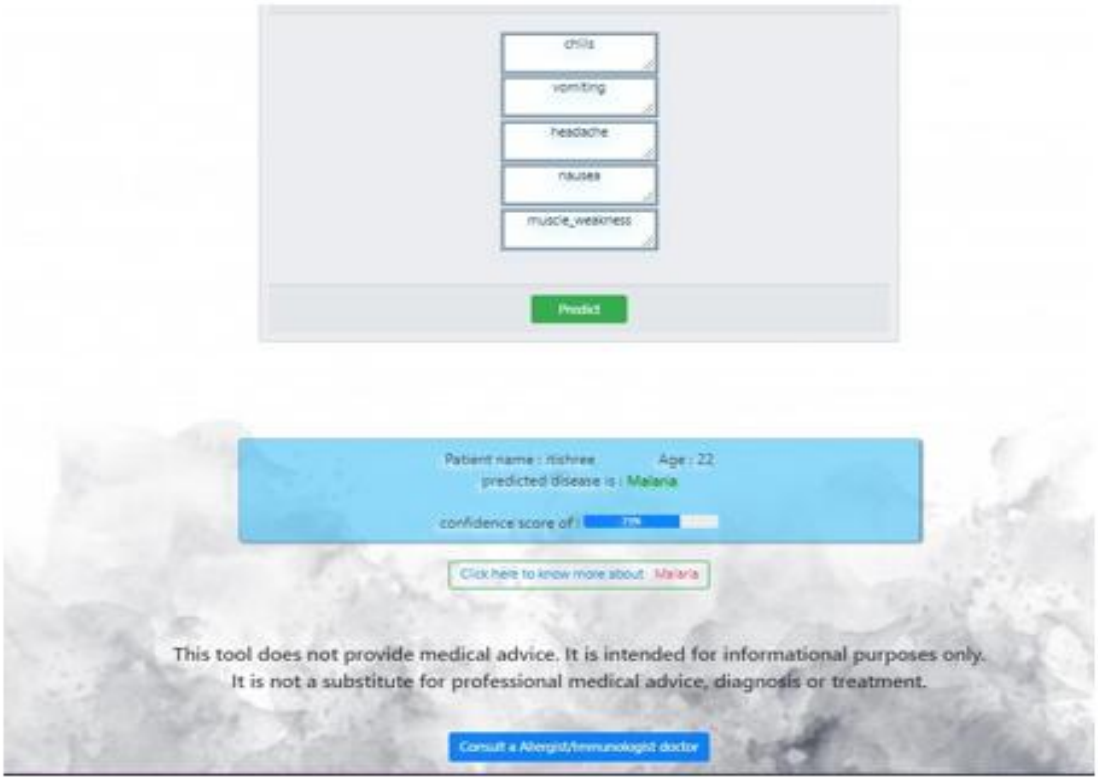
Feedback Form-



Check Disease- Entering symptoms

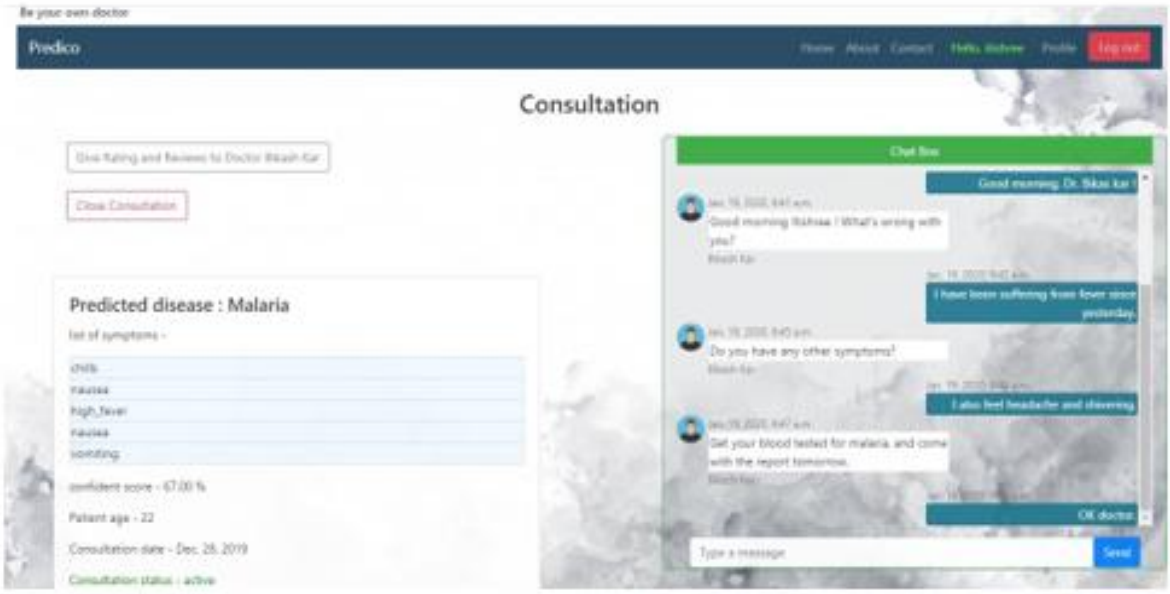


Predictions-

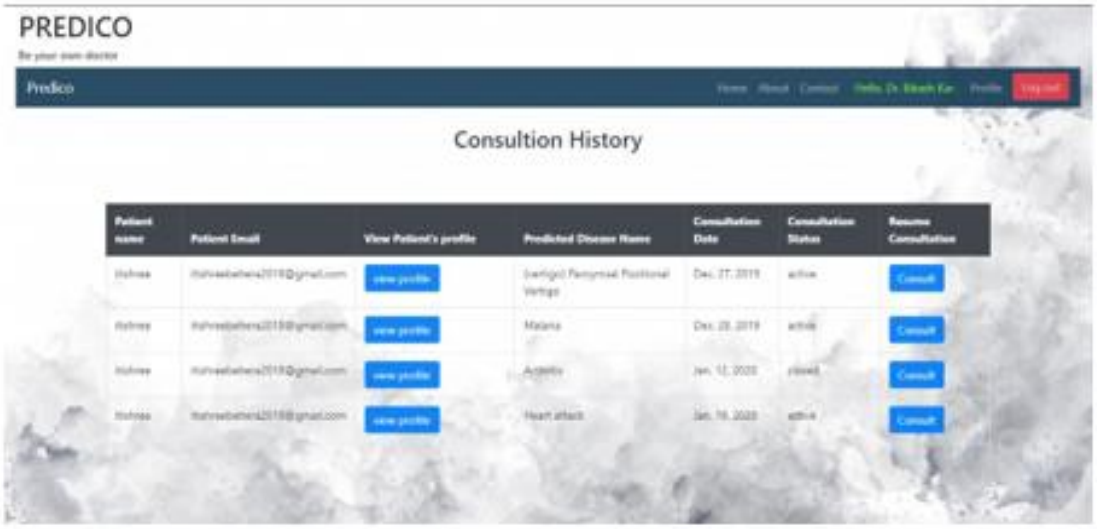


Consult a Doctor-

Consultation UI-

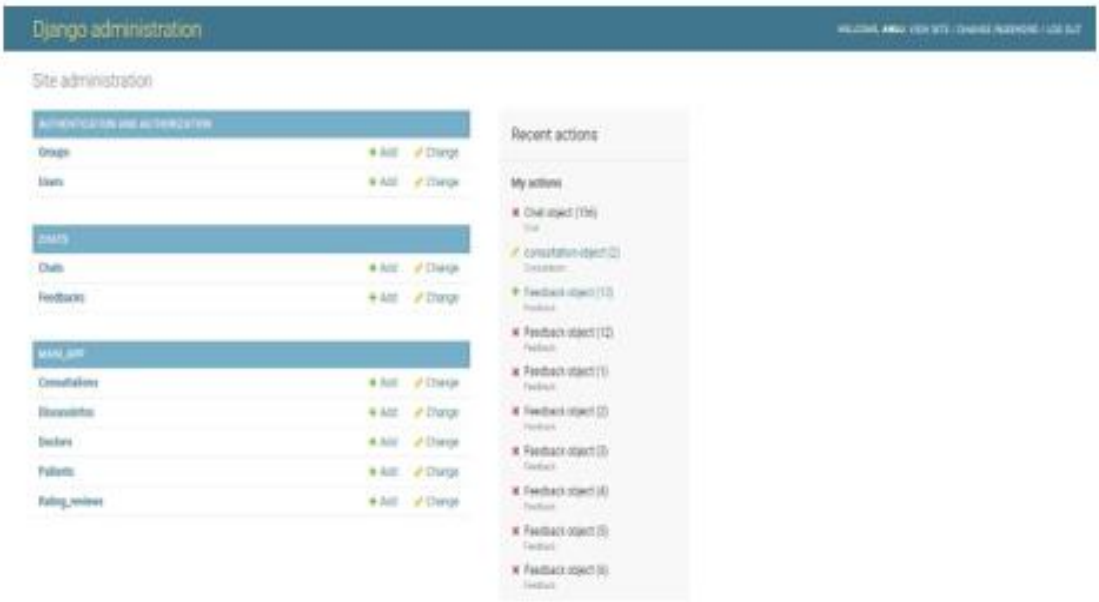


Consultation history- (Doctor)



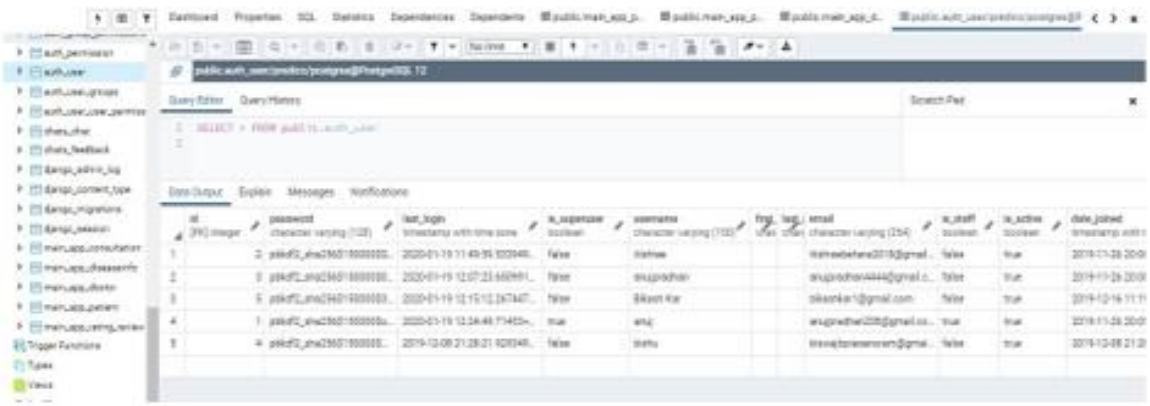
Admin Sign in-

Admin Interface-

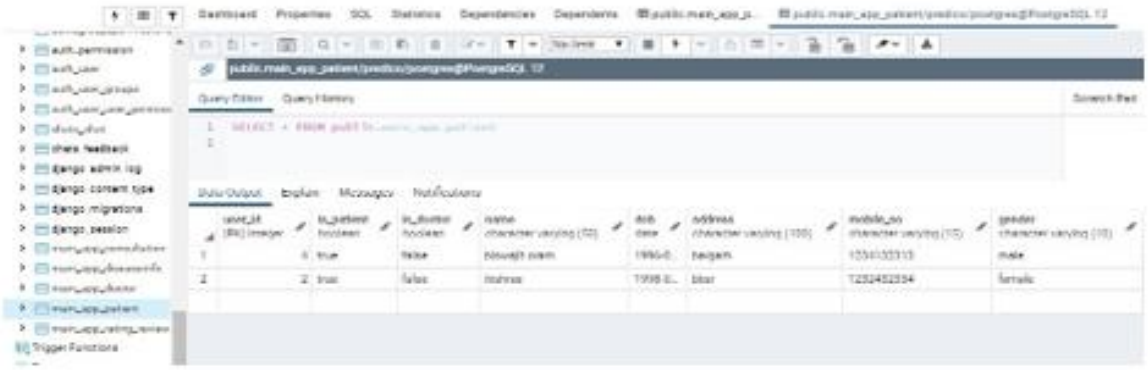


6. Database predico-

Users table-



Patient table-



Consultation table-

	id [PK] integer	consultation_date date	status character	diseaseinfo_id integer	doctor_id integer	patient_id integer
8	34	2019-12-13	active	65	3	2
9	35	2019-12-27	active	67	3	2
10	36	2019-12-27	active	74	5	2
11	37	2019-12-28	active	84	5	2
12	38	2020-01-12	closed	88	5	2
13	2	2019-11-26	active	8	3	2
14	39	2020-01-19	active	93	5	2

10. Challenges Faced:

- Data cleaning and symptom standardization.
- Model performance tuning.
- Integrating ML predictions with a live Django app.
- Designing a secure and scalable database.

11. Learnings:

- Practical implementation of ML in web applications.
- Role-based access control using Django.
- Handling structured medical data.
- Real-time model deployment and integration.

12. Future Scope:

- Addition of **voice-based symptom input.**
- Expansion to include more diseases and multilingual support.
- Integration with **wearables and IoT devices.**
- Collaboration with healthcare institutions for real-world deployment.

13. Conclusion:

- Predico represents a significant leap forward in the digitization of primary healthcare services. By integrating artificial intelligence with an intuitive web interface, the system empowers users to take a proactive approach toward health management. It simplifies the process of disease prediction by allowing users to input symptoms and instantly receive likely diagnoses—reducing dependency on traditional diagnostic cycles that are often time-consuming and resource-intensive.
- The real value of Predico lies not just in its predictive capabilities, but also in how it streamlines the entire healthcare consultation process. Patients are directly connected with appropriate doctors based on the predictions, making the system both responsive and

intelligent. For medical professionals, it offers tools to manage consultations efficiently, ensuring continuity of care through digital records.

- With a well-thought-out architecture and a robust machine learning backbone, Predico sets a strong foundation for future enhancements. It is more than a project—it is a scalable, impactful solution designed to evolve with the growing demands of digital healthcare.

14. References:

This project and its development are supported and inspired by a diverse set of tools, platforms, and educational materials. Below are the references used in crafting the Predico system:

- **Kaggle Dataset**: Disease prediction dataset used for training the machine learning model. Source: Kaggle – Disease Prediction Dataset
- **Django Documentation**: Comprehensive guide to Django's framework features, authentication, and routing systems. Available at: https://docs.djangoproject.com

- **scikit-learn**: The core machine learning library used for model building, training, evaluation, and deployment. Official site: https://scikit-learn.org
- **PostgreSQL**: The open-source relational database used to manage and store user data and consultation records. Reference: https://www.postgresql.org
- **Orange – Data Mining Tool**: Used for visual exploration of data and initial modeling. More information at: https://orange.biolab.si
- **PgAdmin**: A GUI tool for interacting with PostgreSQL, used for database setup and management. Details: https://www.pgadmin.org