Comprehensive Project Report and Next Steps Presentation

Part 1: Comprehensive Project Report

Title: Hypotify Clinical Insights Bot: Database Development and AI Model Justification

Introduction and Domain Justification

The **Hypotify Clinical Insights Bot** project addresses the critical need for efficient data analysis within the **Healthcare** domain. This domain was selected due to the high volume of complex, longitudinal patient data captured in Electronic Medical Records (EMRs) and the stringent ethical requirement for data security. The project utilizes a large-scale **synthetic EMR dataset** (100,000 patients, scaled to 500 for development) to circumvent legal and ethical barriers (HIPAA/PII) associated with accessing real clinical data, allowing for safe development of advanced AI models.

The justification for this domain is rooted in the significant value AI adds through **Predictive Risk Modeling** and **Enhanced Clinical Research** (Rajkomar, Dean, & Kohane, 2019). The bot's ability to quickly query and analyze structured EMR data transforms tedious data extraction into instantaneous insights, accelerating hypothesis testing and clinical decision support.

Relational Database Schema

The core database, **artificial_emr**, was designed using the **MySQL** relational model and rigorously enforced **Third Normal Form (3NF)** to ensure data integrity and query efficiency for the AI components. The schema is organized into four primary entities:

Entity (Table)	Primary Key (PK)	Foreign Keys (FKs)	Rationale for Design

PATIENT	PatientID (VARCHAR(50))	N/A	Stores unique demographics only once (eliminating redundancy).
ADMISSION	AdmissionID (INT)	PatientID	Links patient history to specific encounters (1:M relationship).
DIAGNOSIS	DiagnosisRecordID (INT Auto_Inc)	AdmissionID	Isolates diagnosis codes/descriptions.
LAB_OBSERVATION	LabRecordID (BIGINT Auto_Inc)	AdmissionID	Uses BIGINT to accommodate the 107 million potential records, ensuring scalability and efficient indexing for the ML model.

The decision to use a purely relational model was validated by the data's inherent structure (numerical lab values and categorical demographics), where MySQL's **referential integrity constraints** provide essential trust in the data linkages—a non-negotiable requirement for clinical AI.

Data Preprocessing and Importing Details

The database population culminated in the successful import of the 500-patient dataset into the **artificial_emr** schema (Week 8).

- **Preprocessing:** The primary preprocessing steps focused on data standardization:
    - **Delimiter Check:** Verified all four CSV files used a consistent delimiter to prevent column misalignment during the bulk load.
    - **Date/Time Validation:** Confirmed that all temporal fields (AdmissionStartDate, LabDateTime) were standardized to the exact MySQL YYYY-MM-DD HH:MM:SS format to prevent import failures due to data type mismatch.

- **Importing:** The data was loaded using the MySQL Workbench **Table Data Import Wizard**. The import order was strictly enforced: **PATIENT $\to$ ADMISSION $\to$ DIAGNOSIS $\to$ LAB_OBSERVATION** to satisfy the Foreign Key dependencies.

- **Testing: Accuracy** was verified by checking for zero orphaned records (referential integrity) and **Performance** was tested using a multi-join SQL query (PATIENT $\to$ ADMISSION $\to$ LAB_OBSERVATION for cohort retrieval). The efficient query speed confirmed the success of the 3NF design.

Selection and Justification of Potential AI Models

The chatbot's design relies on a dual-model architecture to handle its specific tasks:

AI Task	Model Selected	Rationale for Suitability
User Interface / Querying	**Text-to-SQL Model** (Fine-Tuned LLM/BERT)	**Justification:** This model translates complex human questions into precise **SQL queries**. It is necessary because the user cannot be expected to know complex table joins. The model's success relies entirely on the **predictable structure** of the 3NF MySQL schema.
Predictive Analysis	**Gradient Boosting Machine (GBM)** (e.g., XGBoost)	**Justification:** GBMs are state-of-the-art for high-accuracy **classification** on structured, tabular data like EMR records. It offers high performance and valuable **Feature Importance scores**, which are critical for providing transparency and interpretability in clinical predictions (Kohn & Alisic, 2018).

Reflection on Experience

The most profound realization from designing and developing this database was the crucial relationship between **data governance** and **AI readiness**. Initially, I viewed normalization as a theoretical exercise,

but the practical difficulties encountered during the Week 8 import (specifically dealing with non-standard date/time formatting) proved its necessity. The experience taught me that the single biggest risk to an AI project is **data inconsistency**.

The most valuable lesson learned is the need for a **Data Staging Pipeline**. In the future, instead of manually cleaning data before import, I will implement transformation scripts (e.g., using Python/Pandas) to automatically validate Foreign Key dependencies and standardize data types *before* any data touches the protected production database. This workflow is essential for minimizing errors, maximizing trust in the data, and ensuring the final ML models are trained on reliable inputs, which is the ethical core of healthcare AI.

Part 2: Next Steps Presentation

The following content is designed for a 3–5 slide recorded presentation, summarizing the ITEC 5025 plan and file management strategy.

Slide 1: Title & Project Summary

- **Title:** Hypotify Clinical Insights Bot: Database Finalization and ITEC 5025 Plan
- **Project Goal:** To develop an AI virtual assistant that translates natural language queries into SQL and performs predictive risk classification on structured EMR data.
- **Foundation:** The fully populated **artificial_emr** database (500-patient structured data, 3NF schema in MySQL).

Slide 2: Summary of Chatbot Development Plan (ITEC 5025)

The ITEC 5025 development phase will be executed in three stages:

1. **Backend Integration (SQLAlchemy):** Use Python to establish a persistent connection to the **artificial_emr** MySQL database via the SQLAlchemy library. This forms the central data access layer.

2. **NLP Pipeline Development (Text-to-SQL):** Build the module that takes user text and generates SQL. This involves fine-tuning a BERT-based model (using Hugging Face) on the artificial_emr schema to ensure accurate query translation.

3. **ML Model Deployment (GBM):** Train the Gradient Boosting Machine (XGBoost) model on the structured lab data to create a high-accuracy risk classification API. The chatbot then calls this API to provide predictive insights to the user.

Slide 3: Next Steps to Prepare for ITEC 5025

To ensure a smooth start in the application development course, these preparatory steps will be completed:

1. **Final Data Validation:** Run comprehensive integrity checks on the **artificial_emr** database one last time (checking all record counts and Foreign Key linkages).

2. **Python Environment Setup:** Install the core development libraries (SQLAlchemy, Pandas, Scikit-learn, XGBoost, and Hugging Face Transformers) on the local machine and verify connectivity to the MySQL database.

3. **Data Extraction Script:** Write the initial Python script that connects to MySQL and pulls the necessary features (LabValue, PatientRace, etc.) into a Pandas DataFrame, preparing the data for immediate ML training.

Slide 4: File Saving and Contingency Plan

It is essential to secure the final database files as they are the primary deliverable for both courses.

- **Where the Database Files are Saved:**
    - **Local Primary:** The database schema and data reside on the MySQL Server instance running on my personal computer.
    - **Cloud Backup 1 (Code & CSVs):** The .sql schema script and the four CSV data files (PatientCorePopulatedTable.csv, etc.) are saved and versioned on **GitHub** (in the repository established in Week 9).
    - **Cloud Backup 2 (Contingency):** A secondary backup of the complete project folder (documentation, CSVs, and SQL script) is stored in **Google Drive/OneDrive**.
- **Contingency Plan:** In the event of a local machine failure, the entire database environment can be instantly recreated on a new machine by downloading the files from **GitHub** and executing the master ITEC5020_Final_Schema_and_Tests.sql script to repopulate the MySQL server.

References

Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of Database Systems* (7th ed.). Pearson.

Kohn, T. A., & Alisic, M. A. (2018). Machine learning methods for analyzing clinical and administrative data: A systematic review. *Journal of Biomedical Informatics*, *87*, 136-146. https://doi.org/10.1016/j.jbi.2018.09.006

Murphy, S. N., Weaver, C. A., & Mendis, M. (2019). Electronic health records and clinical data warehouses. In J. H. Holmes (Ed.), *Clinical Research Informatics*. Springer. https://doi.org/10.1007/978-3-319-98779-0_6

Rajkomar, A., Dean, J., & Kohane, I. (2019). Machine learning in health care. *The New England Journal of Medicine*, *380*(14), 1347–1358. https://doi.org/10.1056/NEJMra1814259

Zhang, Y., Qiu, M., Tsai, C., Hassan, M. M., & Alamri, A. (2017). Health-CPS: Healthcare Cyber-Physical System Assisted by Cloud and Big Data. *IEEE Systems Journal*, *11*(1), 88–95. https://doi.org/10.1109/JSYST.2015.2422055