1. Final Year Project Roadmap

* 🧭 Phase 1: Understanding the Problem & Planning
* Understand the ATIS dataset (Airline Travel Information System):
  + Identify intents (e.g., flight, airfare, ground\_service) and entities (e.g., locations, dates).
  + Review dataset size, format (utterance + intent), and label distribution.
* Define intents and actions your chatbot will support.
* Finalize tech stack (Python, TensorFlow/Keras or PyTorch, scikit-learn, Flask/Streamlit, etc.)
* 🔍 Phase 2: Data Preparation and Preprocessing

1. Load and explore the ATIS dataset.
2. Preprocess text data:
   * Lowercasing, punctuation removal, stopword removal.
   * Tokenization and padding.
   * Label encoding for intents.
3. Split the dataset:
   * Training (80%), Validation (10%), Testing (10%).

* 🧠 Phase 3: Model Development (Intent Recognition)

1. Baseline Model: Start with traditional models:
   * Logistic Regression, SVM, Random Forest (for benchmarking).
2. Deep Learning Model:
   * LSTM/GRU for sequence modeling.
   * Alternatively, use BERT (e.g., bert-base-uncased) for fine-tuning on ATIS.
3. Evaluate performance:
   * Accuracy, Precision, Recall, F1-score, Confusion Matrix.
   * Visualize misclassified intents.

* 😊 Phase 4: Sentiment Analysis Integration

1. Use pre-trained sentiment models (like VADER, TextBlob) or train your own (optional).
2. Classify sentiment of user queries (positive, negative, neutral).
3. Combine output:
   * Intent → maps to action.
   * Sentiment → tunes response tone (empathetic, cheerful, apologetic, etc.)

* 🤖 Phase 5: Chatbot Development (Frontend + Backend)

1. Frontend Interface:
   * Streamlit or simple Angular/React SPA.
   * Chat UI with textbox + response area.
2. Backend Logic:
   * Flask/FastAPI backend.
   * On message → preprocess → run intent model → get intent + sentiment → generate response.
3. Response Generator:
   * Static responses per intent.
   * Add variability based on sentiment.

* 🌐 Phase 6: Deployment

1. Package your model using joblib, pickle, or save as .h5 (Keras).
2. Host API:
   * Flask app on Render, Heroku, or HuggingFace Spaces.
3. Connect frontend to backend using REST API calls.
4. Test real-time interaction.

* 📊 Phase 7: Evaluation & Reporting
* Evaluate chatbot accuracy (intent + sentiment).
* Log user interactions (anonymized) to study behavior.
* Generate insights into:
  + Common misclassifications.
  + User sentiment trends.
  + Frequent intent types.
* 📄 Phase 8: Documentation & Final Report
* Document:
  + Preprocessing steps.
  + Model architecture + hyperparameters.
  + Evaluation metrics + plots.
  + Screenshots of the chatbot interface.
* Include:
  + Ethical considerations & privacy measures.
  + Limitations and future work.
* Create a step-by-step appendix/tutorial for replicability.

1. 🔧 Tools & Libraries

* NLP: NLTK, spaCy, HuggingFace Transformers
* Modeling: TensorFlow/Keras or PyTorch
* Sentiment: VADER, TextBlob
* Frontend: Streamlit, Angular, or React (optional)
* Backend: Flask or FastAPI
* Deployment: Heroku, Render, or HuggingFace Spaces

Would you like me to help you:

1. Start coding the data preprocessing step using the ATIS dataset?
2. Create a design layout for the Streamlit chatbot interface?
3. Draft a Gantt chart or timeline for your FYP report?

Let me know what you want to tackle first!