| **Week** | **Phase** | **Tasks** | **Deliverables** |
| --- | --- | --- | --- |
| **Week 1: Data & Preprocessing** | **Planning & Data Preparation** | - Define project scope and folder structure- Select dataset (movies/products/jobs)- Analyze dataset for missing values, duplicates- Clean dataset- Encode categorical features, scale numeric columns | - Cleaned dataset- Project folder structure- Preprocessing module ready |
| **Week 1–2: EDA & Recommendation Engine** | **Exploration & Modeling** | - Perform Exploratory Data Analysis (visualizations, summary stats)- Generate AI textual insights for dataset- Build content-based recommendation system- Implement collaborative filtering (user-item matrix)- Optionally combine for hybrid model | - EDA visuals & insights- Functional recommendation engine- Similarity matrices computed |
| **Week 2: Explainable AI & Evaluation** | **Interpretability & Testing** | - Add explainable AI (SHAP/LIME) to recommendations- Evaluate model using metrics: precision, recall, RMSE, etc.- Fine-tune model if necessary | - Recommendation explanations- Model evaluation report- Improved recommendation accuracy |
| **Week 2–3: Dashboard & Interactivity** | **App Development** | - Build interactive Streamlit/Flask dashboard- Allow dataset upload, item selection, feature filters- Display recommendations dynamically- Integrate charts/visualizations | - Interactive dashboard- Recommendations displayed live- User-friendly interface |
| **Week 3: Report & Deployment** | **Reporting & Deployment** | - Generate automated PDF/HTML reports (dataset insights + recommendations)- Deploy app on Streamlit Cloud or Heroku- Test for bugs and user experience | - Deployed project- Downloadable report- Fully functional, portfolio-ready project |

## ****1️⃣ Project Goal****

Build a **fully AI-driven, interactive recommendation system** that:

* Recommends items/products/movies/jobs to users based on data.
* Automatically preprocesses, analyzes, and computes similarity.
* Provides explanations for recommendations.
* Generates downloadable reports.
* Can be deployed as an interactive **dashboard/app**.

## ****2️⃣ Target Users****

* Recruiters or data science enthusiasts analyzing datasets.
* Businesses wanting product or content recommendations.
* Students/projects demonstrating end-to-end ML workflow.

## ****3️⃣ Key Features****

1. **Data Ingestion**
   1. Upload CSV/Excel datasets or connect via API.
   2. Support for multiple types of data: movies, products, jobs.
2. **Data Preprocessing (Automated)**
   1. Handle missing values and duplicates.
   2. Encode categorical features automatically.
   3. Normalize/scale numerical data.
3. **Exploratory Data Analysis (EDA)**
4. Show dataset summary: shape, missing values, distribution.
5. Visualizations: histograms, correlation heatmaps.
6. Optional AI-generated textual insights.
7. **Recommendation Engine**
8. **Content-based filtering:** Use item features to recommend similar items.
9. **Collaborative filtering:** Use user ratings/behavior to suggest items.
10. **Hybrid approach:** Combine both for better accuracy.
11. Predict unknown ratings or preferences using AI/ML.
12. **Explainable Recommendations**
13. Show reasons for each recommendation (feature influence, similarity score).
14. Optional: use SHAP/LIME to visualize importance of features.
15. **Interactive Dashboard**
16. Users can select an item and see recommendations.
17. Filter recommendations based on categories, ratings, or other criteria.
18. **Report Generation**
19. Generate a PDF report summarizing:
    1. Dataset insights
    2. Recommendations
    3. Model evaluation (if applicable)
20. **Evaluation Metrics**
    1. Precision, recall, F1-score for rating predictions.
    2. Mean squared error (MSE) or RMSE for numerical ratings.

## ****5️⃣ Tech Stack****

* **Python Libraries:** pandas, numpy, scikit-learn, surprise, scipy, matplotlib, seaborn, plotly
* **AI/ML Tools:** content & collaborative filtering, AutoML (optional)
* **Explainability:** SHAP or LIME
* **Deployment:** Streamlit / Flask
* **Reporting:** FPDF or pdfkit for PDFs

| **Phase** | **Tasks** | **Outcome** |
| --- | --- | --- |
| **Phase 1: Planning** | Define project scope, folder structure, dataset selection, tools | Clear roadmap |
| **Phase 2: Data Collection** | Gather datasets (movies/products/jobs), clean & explore | Ready-to-use data |
| **Phase 3: Preprocessing** | Handle missing values, encoding, scaling | Clean dataset |
| **Phase 4: EDA** | Visualize data, generate insights | Understanding of dataset |
| **Phase 5: Recommendation Engine** | Implement content-based, collaborative, hybrid filtering | Functional model |
| **Phase 6: Explainable AI** | Show reasons for recommendations | Transparent model |
| **Phase 7: Dashboard/App** | Build Streamlit/Flask interface | Interactive user experience |
| **Phase 8: Report Generation** | Auto-generate PDF with insights & recommendations | Professional output |
| **Phase 9: Evaluation & Testing** | Evaluate recommendations, metrics | Reliable system |
| **Phase 10: Deployment** | Deploy online (Streamlit Cloud / Heroku) | Public portfolio project |

**Optional Enhancements**

* **Chatbot Integration** – Users can ask for recommendations via natural language.
* **AI Insights** – Textual insights summarizing top features or trends.
* **Dynamic Filters** – Allow filtering recommendations by rating, category, or other features.
* **AutoML Predictions** – Predict missing ratings or preferences automatically.

## ****Tips to Make it Stand Out for Job Applications****

1. **End-to-End Demonstration**: Showcase everything from raw data → preprocessing → recommendations → report → deployment.
2. **Interactivity**: Recruiters love projects they can click through.
3. **Explainable AI**: Adding SHAP/LIME shows advanced skills.
4. **Deployment**: A live URL in your resume or LinkedIn portfolio adds credibility.
5. **Documentation**: Include README, screenshots, and short video demo if possible.

Week 1 Week 2 Week 3

|----------------|----------------|----------------|

Phase 1: Planning & Data Prep

[██████████] - Define scope & folder structure

[██████████] - Dataset selection & cleaning

[██████████] - Preprocessing module ready

Phase 2: EDA & Recommendation Engine

[████████████] - Exploratory Data Analysis

[████████████] - AI textual insights

[████████████] - Content-based recommendation

[████████████] - Collaborative filtering

[████████] - Hybrid model implementation

Phase 3: Explainable AI & Evaluation

[████████] - SHAP/LIME explainability

[████████] - Model evaluation & tuning

Phase 4: Dashboard & Interactivity

[████████████] - Streamlit/Flask dashboard

[████████████] - Interactive recommendations

[██████████] - Charts & visualizations

Phase 5: Report & Deployment

[██████████] - PDF/HTML report generation

[██████████] - Deployment & testing

+---------------------------------------------------+

| STREAMLIT APP |

+---------------------------------------------------+

| Sidebar: Navigation |

| - Home |

| - Dataset |

| - Model |

| - Recommendation |

+---------------------------------------------------+

| Dataset Page |

+---------------------------------------------------+

| Step 1: User selects data type (Tab / Text) |

| |

| Step 2: User uploads dataset |

| - Tabular Data (CSV / Excel) |

| - Sequential/Text Data (TXT / LOG / JSON) |

+---------------------------------------------------+

| DATA PREPROCESSING |

+---------------------------------------------------+

| Tabular Data: |

| - Missing Value Handling |

| - Categorical Encoding |

| - Scaling / Normalization |

| - Outlier Handling (Optional) |

| - Feature Selection (Optional) |

| - Output: df\_processed, scaler, encoder\_dict |

| |

| Text Data: |

| - Text Cleaning (lowercase, remove punctuation) |

| - Tokenization |

| - Vectorization (TF-IDF / CountVectorizer) |

| - Optional Dimensionality Reduction (PCA/SVD) |

| - Output: text\_matrix, vectorizer |

+---------------------------------------------------+

| MODEL PAGE |

+---------------------------------------------------+

| Step 3: User selects algorithm |

| - Content-Based |

| - Collaborative Filtering |

| - Hybrid |

| Step 4: Train & Evaluate model |

+---------------------------------------------------+

| RECOMMENDATION PAGE |

+---------------------------------------------------+

| Step 5: User enters item name / query |

| Step 6: Generate Recommendations |

| - Use preprocessed Tabular or Text data |

| - Output top-N recommended items |

+---------------------------------------------------+

| SESSION\_STATE |

+---------------------------------------------------+

| Stores uploaded and preprocessed datasets: |

| - uploaded\_tabular / uploaded\_text |

| - df\_processed / text\_matrix |

| - scaler / vectorizer / encoder\_dict |

| - text\_column if text data |

+---------------------------------------------------+