**1. Problem Definition & Goal Setting**

**Objective:**

Build a **Product Recommendation System** that suggests products to users based on their preferences, behavior, and purchase history.

**Types of Recommendation Systems:**

* **Collaborative Filtering:** Uses user-item interactions (e.g., purchases, ratings).
* **Content-Based Filtering:** Recommends similar items based on item features.
* **Hybrid Approach:** Combines both methods for better accuracy.

**Technologies Used:**  
📌 **Python**, **Jupyter Notebook**, **MS Excel** (for initial data exploration)

**2. Data Collection & Storage**

**Datasets:**

* **User Data:** User ID, age, gender, location, preferences
* **Product Data:** Product ID, category, description, price
* **Interaction Data:** Ratings, clicks, purchase history

**Data Sources:**

* **E-commerce Websites (Web Scraping)**
* **Public Datasets (Kaggle, UCI, etc.)**
* **Company’s Internal Database**

**Technologies Used:**  
📌 **Python (BeautifulSoup, Scrapy)** – Web scraping  
📌 **APIs (Amazon Product API, Flipkart API)**  
📌 **SQL / NoSQL (MySQL, PostgreSQL, MongoDB)** – Data storage

**3. Data Preprocessing & Cleaning**

* Handling **missing values**
* Removing **duplicates and irrelevant data**
* **Feature engineering** (creating new features like user engagement score)
* Encoding **categorical variables**

**Technologies Used:**  
📌 **Python (Pandas, NumPy, Scikit-learn)**

**4. Exploratory Data Analysis (EDA)**

* Understanding data distribution (histograms, boxplots)
* Finding correlations (heatmaps)
* Analyzing user behavior trends

**Technologies Used:**  
📌 **Python (Matplotlib, Seaborn, Power BI/Tableau for visualization)**

**5. Model Selection & Training**

**Approach 1: Collaborative Filtering**

* **User-based Filtering:** Users with similar tastes
* **Item-based Filtering:** Items that are frequently bought together
* **Model Used:** Matrix Factorization (SVD, ALS)

**Approach 2: Content-Based Filtering**

* Uses **TF-IDF, Word2Vec, or BERT** for text-based recommendations
* **Cosine Similarity, Euclidean Distance** to find similar items

**Approach 3: Deep Learning-based Hybrid Approach**

* **Neural Collaborative Filtering (NCF)**
* **Autoencoders for latent representation learning**

**Technologies Used:**  
📌 **Scikit-learn, Surprise Library, TensorFlow/Keras, PyTorch**

**6. Model Evaluation**

* **Accuracy Metrics:** RMSE, Precision@K, Recall@K
* **A/B Testing** to compare multiple models

**Technologies Used:**  
📌 **Python (Scikit-learn, MLflow for tracking experiments)**

**7. Building the Backend API**

* **Creating APIs** for recommendations
* **Storing results in a database**

**Technologies Used:**  
📌 **FastAPI / Flask / Django** (for REST API)  
📌 **PostgreSQL / MongoDB** (for storing recommendations)

**8. Frontend Development**

* **UI Design:** Product catalog, recommended products section
* **User Authentication & Dashboard**

**Technologies Used:**  
📌 **React.js / Next.js / Angular.js**  
📌 **TailwindCSS / Bootstrap** (for styling)

**9. Model Deployment**

* **Deploy API on Cloud** (AWS, Azure, Google Cloud)
* **Use Docker for containerization**
* **Deploy Frontend on Vercel/Netlify**

**Technologies Used:**  
📌 **Docker, Kubernetes, AWS Lambda, Firebase**

**10. Monitoring & Optimization**

* Tracking **user feedback & engagement**
* Improving **recommendations using reinforcement learning**

**Technologies Used:**  
📌 **Prometheus & Grafana (for monitoring)**  
📌 **Google Analytics (for tracking user behavior)**

**Final Tech Stack Summary**

| **Phase** | **Technology** |
| --- | --- |
| **Data Collection** | Python, Web Scraping, SQL, MongoDB |
| **EDA** | Pandas, Matplotlib, Seaborn, Power BI |
| **Modeling** | Scikit-learn, TensorFlow, PyTorch |
| **Backend** | Flask / FastAPI, PostgreSQL |
| **Frontend** | React.js, TailwindCSS |
| **Deployment** | AWS, Docker, Kubernetes |