

---

# Complete Project Chunk Breakdown (Up to Phase 3)

(Balanced between database, backend, frontend, ML, and optional MLOps layers)

---

## CHUNK 1 – Setup & Team Onboarding

**Goal:** Prepare everything to start development smoothly.

### Tasks

- Approve final project scope (till Phase 3).
  - Assign roles: DB Lead, Backend Lead, Frontend Lead, ML Lead, MLOps/Deployment Lead.
  - Setup GitHub repo with folders:  
`/backend, /frontend, /database, /ml, /data, /docs`
  - Setup coding conventions, version control workflow (branching + pull requests).
  - Collect datasets (H&M, RetailRocket, Amazon Reviews, DeepFashion).
  - Generate synthetic data (Mockaroo/Faker).
  - Tools: GitHub, Notion, draw.io, VSCode, Python, MySQL/Postgres.
- 

## CHUNK 2 – Database System (Core DB Foundation)

**Goal:** Design and implement your e-commerce database (BCNF).

### Tasks

- Finalize schema (Customer, Product, Category, Order, OrderDetails, Review, Campaign, Influencer).
- Draw ERD and write normalization documentation ( $1\text{NF} \rightarrow \text{BCNF}$ ).
- Write SQL scripts:
  - `create_tables.sql, insert_data.sql, create_views.sql`
- Add stored procedures:
  - Order placement (transaction)

- Admin reports
  - Example views:
    - `top_selling_products`, `low_stock_items`, `daily_revenue`
  - Validate using sample data and ensure CRUD operations.
- Deliverables**
- Database ready + SQL scripts + ERD + screenshots.
- 

## CHUNK 3 – Backend API (FastAPI + Core Business Logic)

**Goal:** Build backend that connects DB + ML + frontend.

### Tasks

- Initialize FastAPI project.
- Create endpoints:
  - `/products`, `/orders`, `/reviews`, `/recommendations`, `/forecasts`
- Connect to database using SQLAlchemy.
- Implement order placement logic with stock update & transactions.
- Add basic authentication (JWT or session-based).
- Return JSON responses to frontend.

**Deliverables**

- Working backend API + Postman test collection.
- 

## CHUNK 4 – Frontend (Website Interface)

**Goal:** Build full working website for customers and admins.

### Tasks

- Choose tech: React or Next.js (preferred), or Streamlit prototype for speed.
- Implement pages:
  - **Customer:** Browse products, add to cart, checkout, view recommendations.
  - **Admin:** Dashboard (inventory, sales reports, trends, segmentation).

- Connect with FastAPI backend (Axios / Fetch).
- Add chart visualizations (Chart.js / Recharts).

### Deliverables

- Frontend + backend integrated website running locally.
- 

## CHUNK 5 – ETL + Data Engineering

**Goal:** Enable data flow from raw CSVs → DB → ML.

### Tasks

- ETL scripts using Python (pandas):
  - Extract Kaggle data, clean, transform, and load into DB.
- Compute:
  - RFM features, sales history, category-level aggregates.
- Store cleaned tables for ML (e.g., `customer_product_matrix.csv`).
- Optional: Prefect / Airflow DAG for automated ETL.

### Deliverables

- Clean database populated with meaningful, ML-ready data.
- 

## CHUNK 6 – Core ML Models (MVP)

**Goal:** Add intelligence – recommendations + forecasting.

### Models

#### 1. Collaborative Filtering Recommender (ALS / Matrix Factorization)

Input: customer-product interactions

Output: Top-N recommendations

#### 2. Sales Forecasting (Prophet / ARIMA)

Input: product/category sales history

Output: next-period sales predictions

### Integration

- Serve via FastAPI routes `/recommendations` & `/forecast`.
- Store outputs in DB or cache layer.

### Deliverables

- ML notebook + integrated backend endpoints + visual demo on frontend.

## CHUNK 7 – Phase 2 Enrichment (Hybrid & Segmentation)

**Goal:** Make ML more insightful and business-driven.

### Tasks

- **Hybrid Recommender:** combine collaborative + product content (category, price, embeddings).
- **Customer Segmentation:** KMeans/GMM on RFM & product interactions.
- Add “Segment Insights” dashboard in admin UI.
- Update backend to serve segmentation API.

### Deliverables

- Notebooks + visual reports + integration in admin panel.

---

## CHUNK 8 – Phase 3 Analytics (Sentiment & Trends)

**Goal:** Add advanced insights & analytics.

### Tasks

- **Sentiment Analysis on Reviews:**
  - Model: VADER (quick) or BERT (deep)
  - Output: sentiment scores per product/category.
- **Trend Prediction:**
  - Combine sentiment + sales velocity + Google Trends.
  - Output: trending categories/styles dashboard.
- Integrate into admin dashboard (charts).

### Deliverables

- Sentiment notebook, trend notebook, updated frontend panel.

---

## CHUNK 9 – MLOps Layer (Optional Stretch Goals)

**Goal:** Add professional tools if time allows.

### Priority Order:

1. **Docker:** containerize backend + ML model APIs.
2. **MLflow:** log model metrics, parameters, and version models.
3. **Airflow:** schedule ETL & retraining pipelines.
4. **CI/CD (GitHub Actions):** automate testing, build, and deploy.

5. **EvidentlyAI**: track data drift (if time).

6. **Monitoring (Prometheus + Grafana)**: track API uptime & model health.

#### Deliverables

- Containers + MLflow dashboard + optional Airflow DAGs.
- 

## CHUNK 10 – Final Integration, Testing & Deployment

**Goal:** Package everything professionally.

#### Tasks

- End-to-end integration (DB ↔ FastAPI ↔ Frontend ↔ ML).
  - Containerize backend + ML APIs using Docker Compose.
  - Deploy to cloud (Render / AWS / Railway).
  - Add sample users and data.
  - Write documentation & prepare final report + demo video.
- 

## 🧭 Final Order of Priority (If Time Becomes Tight)

Priority	Must-Have	Nice-to-Have (If Time Allows)
✓ 1	Database (schema, CRUD, transactions)	—
✓ 2	Backend API (FastAPI, DB integration)	Authentication, pagination
✓ 3	Frontend website	Admin dashboards
✓ 4	Core ML (Recommender + Forecasting)	More models
✓ 5	Sentiment + Trend analytics	Social data integration
🟡 6	Docker containerization	—
🟡 7	MLflow tracking	Airflow, CI/CD
🟢 8	Airflow, CI/CD, EvidentlyAI, Monitoring	Production-grade scaling

---