# Customer Churn Dashboard — Day-wise 3-Month Schedule

This document provides a **day-by-day schedule for 12 weeks (3 months)** to implement a Customer Churn Prediction Dashboard, including preparation, optional steps, and deliverables.

## Week 1 — Project Setup & Discovery

| Day | Task | Deliverables |
| --- | --- | --- |
| 1 | Create GitHub repo, initialize project structure | GitHub repo, folder skeleton, README.md stub |
| 2 | Define problem, success metrics, KPI targets | Problem statement doc, KPI doc |
| 3 | Collect datasets (Kaggle Telco / optional) | Raw datasets in data/raw/ |
| 4 | Set up Python virtual env & install packages | venv setup, requirements.txt |
| 5 | Churn definition finalized & documented | docs/churn\_definition.md |

## Week 2 — Data Ingestion & EDA

| Day | Task | Deliverables |
| --- | --- | --- |
| 6 | Load dataset, initial inspection | notebooks/01\_EDA.ipynb basic info |
| 7 | Visualize churn distribution & correlations | EDA plots, insights summary |
| 8 | Label creation pipeline | src/features/label.py |
| 9 | Data validation scripts | src/data/validate.py, unit tests |
| 10 | Feature catalog drafted | docs/feature\_catalog.md |

## Week 3 — Feature Engineering & Preprocessing

| Day | Task | Deliverables |
| --- | --- | --- |
| 11 | Implement numerical & categorical feature transformations | src/features/build\_features.py |
| 12 | Create time-window features | data/processed/train.parquet |
| 13 | Save feature metadata | JSON metadata file |
| 14 | Unit tests for feature functions | tests/test\_features.py |
| 15 | Run end-to-end pipeline locally | Processed dataset validated |

## Week 4 — Baseline Modeling & Validation

| Day | Task | Deliverables |
| --- | --- | --- |
| 16 | Time-based train/validation split | Split datasets for modeling |
| 17 | Train logistic regression baseline | Initial model artifacts |
| 18 | Train XGBoost / tree-based model | models/xgb\_churn.joblib |
| 19 | Evaluate metrics (AUC, PR-AUC, Precision@K) | notebooks/02\_model\_eval.ipynb |
| 20 | Baseline model card documented | docs/model\_card\_baseline.md |

## Week 5 — Advanced Modeling & Explainability

| Day | Task | Deliverables |
| --- | --- | --- |
| 21 | Hyperparameter tuning & final model | Final trained model |
| 22 | Probability calibration | Calibration saved if applicable |
| 23 | SHAP global & per-customer analysis | SHAP explainer & plots |
| 24 | Create script to return top-N features | src/model/explain.py |
| 25 | Model evaluation notebook final version | notebooks/03\_final\_model\_eval.ipynb |

## Week 6 — Model Packaging & Testing

| Day | Task | Deliverables |
| --- | --- | --- |
| 26 | Wrap model into callable class | src/model/predictor.py |
| 27 | CLI to score CSV | scripts/score\_csv.py |
| 28 | Unit tests for predictor | tests/test\_predictor.py |
| 29 | CI/CD dev setup, linting | requirements-dev.txt, tox config |
| 30 | Run tests & fix issues | All unit tests pass |

## Week 7 — FastAPI Model Serving

| Day | Task | Deliverables |
| --- | --- | --- |
| 31 | FastAPI skeleton setup | src/api/app.py basic endpoints |
| 32 | Implement /predict & /batch\_predict | JSON response with churn + top features |
| 33 | Input validation & error handling | Pydantic models implemented |
| 34 | Run local API server | uvicorn running at localhost:8000 |
| 35 | Logging & basic rate limiting | Logs for API requests, throttle checks |

## Week 8 — Containerization & Batch Scoring

| Day | Task | Deliverables |
| --- | --- | --- |
| 36 | Dockerfile for API | docker/Dockerfile.api |
| 37 | Build & run Docker container locally | Containerized API working |
| 38 | Airflow DAG skeleton for batch scoring | airflow/dags/batch\_score.py |
| 39 | CI workflow for Docker image | GitHub Actions config |
| 40 | Test batch scoring | Predictions file generated nightly |

## Week 9 — Dashboard UI (Streamlit) MVP

| Day | Task | Deliverables |
| --- | --- | --- |
| 41 | Sidebar filters, KPI boxes | src/dashboard/app.py skeleton |
| 42 | Customer table & CSV export | Table displays top customers, export works |
| 43 | Per-customer detail & SHAP plots | Waterfall or summary plots integrated |
| 44 | Basic authentication | Simple password or secrets-based auth |
| 45 | Local run & screenshots | Screenshots saved in docs/screenshots/ |

## Week 10 — Tests, Security & Privacy

| Day | Task | Deliverables |
| --- | --- | --- |
| 46 | Mask PII in UI | Customer identifiers anonymized |
| 47 | Move secrets out of code | .env or st.secrets configured |
| 48 | Unit/integration tests for API & dashboard | Tests passing via pytest |
| 49 | Privacy & security docs | docs/privacy.md, docs/security.md |
| 50 | Security checklist run | HTTPS, logging, role separation verified |

## Week 11 — Deployment & CI/CD

| Day | Task | Deliverables |
| --- | --- | --- |
| 51 | GitHub Actions: tests & Docker | CI pipeline functional |
| 52 | Deploy API to Cloud / local | API running live or ngrok link |
| 53 | Deploy Streamlit app | Dashboard accessible via URL |
| 54 | Integration check | Dashboard calls API or reads predictions table |
| 55 | Smoke tests & UAT | All features verified |

## Week 12 — Polish, Documentation & Portfolio Prep

| Day | Task | Deliverables |
| --- | --- | --- |
| 56 | Finalize README | README.md with screenshots, quickstart, and demo links |
| 57 | Record demo video (2–3 mins) | Video hosted (YouTube / Drive) |
| 58 | Create case study PDF | docs/case\_study.pdf 1-page summary |
| 59 | Prepare interview answers & talking points | Documented Q&A + elevator pitch |
| 60 | Publish repo & LinkedIn announcement | Repo tagged v1.0, post live |

### Notes:

* Optional / extra tasks (feature store, MLflow, LIME, A/B testing) can be inserted if you have extra time.
* Each week is designed to finish Friday with deliverables ready for review.
* Commands for running on Linux, Docker, Streamlit, and FastAPI should be integrated as described in the roadmap.

This schedule ensures **no step is missed**, from environment setup to portfolio-ready deployment.

**End of Day-wise 3-Month Schedule**