Make a detailed HTML website about quantitative finance and its rich history with the following pages

1. **Home: What is Quant?**
    - Plain-language definition vs. “finance” broadly; what makes it *quantitative* (models, data, automation).
    - Diagram of the quant pipeline: **data** → **hypothesis** → **model** → **backtest** → **risk** → **execution** → **monitoring**.
    - **Research task:** contrast discretionary PM vs. systematic/quant; 2–3 bullet differences with sources.
2. **History & Landscape**
    - Timeline (1950s–today): Markowitz → Black-Scholes → statistical arbitrage → HFT → ML/alt-data.
    - Roles: **quant researcher, strat, risk quant, execution/algo, data engineer, quant dev**—what each actually ships.
    - **Research task:** add 4–6 dated milestones; cite primary sources (original papers/official exchange docs where possible).
3. **Core Math & Methods**
    - Short explainers (with equations via KaTeX/MathJax): **linear algebra**, **stochastic processes (Brownian motion, Ito)**, **optimization (mean–variance, convex)**, **time series (ARIMA, GARCH)**, **ML in quant (trees, boosting, regularization) — strengths/limits**.
    - **Research task:** for each method, include **assumptions**, **failure modes**, and a **canonical reference**.
4. **Strategies & Evaluation**
    - Strategy archetypes: **trend/momentum, mean reversion/stat-arb, factor investing (value/size/quality), volatility trading, market-making, event-driven**, plus **portfolio construction**.
    - Evaluation: **backtesting discipline**, **walk-forward**, **cross-validation for time series**, **transaction costs**, **turnover**, **risk metrics** (vol, drawdown, Sharpe, Sortino, information ratio, max DD), **p-hacking & data-snooping**.
    - **Research task:** pick **one strategy** and present:
        - clear hypothesis,
        - realistic cost model,
        - pitfalls (look-ahead bias, survivorship bias),
        - at least **2 academic/industry references** that agree or disagree.
5. **Careers & Tooling** (optional but encouraged)
    - Tech stack: **Python (NumPy/Pandas), C++/Java for low-latency, SQL, KDB/q**, job ladders, interview focuses.
    - **Research task:** compare “quant researcher” vs “quant developer” job postings—skills overlap vs differences.