Shared Budgeting App: Full Architecture and Step-by-Step Plan

1. Full Model Architecture

Entities & Data Models:

- User: id, name, email, password_hash, created_at
- Household: id, name, created_at
- HouseholdUser: household_id, user_id, role (owner/admin/member)
- Account: id, household_id, name, type (bank/card), last4, currency
- Transaction: id, account_id, date, description, amount, category_id, user_id, source_file_id
- Category: id, name, parent_id, default_budget
- Income: id, user_id, date, amount, source, notes
- Goal: id, user_id, name, target_amount, current_amount, target_date, recurrence
- File: id, household_id, user_id, s3_key, parsed_json_key, status

High-Level System Components:

- Frontend UI: React + Tailwind CSS, dashboard, forms, file upload, analytics.
- Backend API: FastAPI, authentication, file upload, parsing orchestration, CRUD.
- File Storage: AWS S3 or local dev storage.
- Database: PostgreSQL (prod), SQLite (dev).
- Parsing Worker: Async jobs for PDF parsing, OCR, normalization.
- ML Classification: Auto categorization and anomaly detection.
- Analytics Engine: Aggregations and suggestions.
- Authentication: OAuth/JWT, household sharing.

2. What is Already Done

- React dashboard mockup with combined views and transaction layout.
- FastAPI backend starter with /upload-statement endpoint for raw PDF text extraction.

- Basic database model skeleton (users, transactions).

3. Next Steps: Detailed Plan to Build

Step 1: Set Up Database & Authentication

- Implement full DB schema, user registration/login with JWT.
- Household sharing with roles and invites.

Step 2: File Upload & Parsing Pipeline

- Extend file type support, async job queue for parsing.
- PDF parsing with pdfplumber and OCR fallback.
- Normalize transactions and store parsed data.

Step 3: Categorization & Transaction Management

- Rule-based and ML-based categorization.
- CRUD for transactions and user assignments.

Step 4: Income & Expenses Tracking

- Income entry, automatic balance calculation, historical trends.

Step 5: Goals & Investment Tracking

- Create and track savings/investment goals with progress visualization.

Step 6: Analytics & Insights

- Aggregated spending data, alerts, suggestions.

Step 7: Polish, Security & Deployment

- Add security measures, CI/CD, deploy backend/frontend.

4. Timeline Suggestion

Week 1: Database, Authentication, Household Sharing

Shared Budgeting App - Architecture & Development Plan

Week 2: File Upload & Parsing Worker

Week 3: Categorization & Transactions CRUD

Week 4: Income & Expenses Tracking UI

Week 5: Goals Tracking & Progress UI

Week 6: Analytics, Alerts, Suggestions

Week 7: Testing, Security, Deployment

5. Tools & Libraries

- Frontend: React, Tailwind CSS, Chart.js/Recharts
- Backend: FastAPI, SQLAlchemy, Alembic, RQ/Celery, pdfplumber, pytesseract
- Database: PostgreSQL (production), SQLite (development)
- Storage: AWS S3 or local
- ML: Python scikit-learn or LightGBM
- Authentication: OAuth2 / JWT