Отлично! Составлю пошаговый план на 7 дней с акцентом на практику. Учтём ваши базовые знания и необходимость создать MVP интернет-магазина.

**День 1: Настройка окружения и базовый бэкенд**

1. **Установка:**
   * PostgreSQL: создание БД online\_shop через pgAdmin
   * FastAPI: pip install fastapi sqlalchemy asyncpg uvicorn
   * React: npx create-react-app frontend
2. **Бэкенд (FastAPI + SQLAlchemy):**
   * Создайте модели для Product и User:
3. class Product(Base):
4. \_\_tablename\_\_ = "products"
5. id = Column(Integer, primary\_key=True)
6. name = Column(String)
7. price = Column(Float)
8. description = Column(String)
   * Настройте подключение к PostgreSQL через asyncpg
   * Создайте эндпоинт GET /products для получения списка товаров. [[stepik.org](https://stepik.org/course/179694/promo)stepik.org](https://stepik.org/course/179694/promo)

**День 2: Углублённая работа с FastAPI**

1. **Добавьте:**
   * JWT-аутентификацию для /users/me
   * Валидацию данных через Pydantic:
2. class ProductCreate(BaseModel):
3. name: str = Field(..., min\_length=3)
4. price: float = Field(gt=0)
   * Эндпоинты POST/PUT/DELETE для управления товарами
   * Документацию Swagger/Redoc через @app.get("/", include\_in\_schema=False)
5. **Тестирование** API через Postman. [[metanit.com](https://metanit.com/python/fastapi/1.1.php)metanit.com](https://metanit.com/python/fastapi/1.1.php)

**День 3: Базовый фронтенд (React)**

1. **Создайте:**
   * Компонент ProductList с выводом карточек товаров
   * Хук useFetch для запросов к API:
2. const { data, loading } = useFetch('http://localhost:8000/api/products')
   * Контекст CartContext для управления корзиной
3. **Настройте** роутинг через React Router:
4. <Route path="/product/:id" element={<ProductPage />} />
5. **Используйте** Ant Design для быстрого прототипирования. [[sky.pro](https://sky.pro/wiki/javascript/react-dlya-nachinayushih-s-chego-nachat/)sky.pro](https://sky.pro/wiki/javascript/react-dlya-nachinayushih-s-chego-nachat/)

**День 4: Интеграция фронтенда с бэкендом**

1. **Реализуйте:**
   * Загрузку товаров с пагинацией (добавьте skip и limit в API)
   * Поиск по названию через параметр запроса ?q=...
   * Обработку ошибок при 401/404 статусах
2. **Добавьте** формы:
   * Авторизации (JWT токен в localStorage)
   * Добавления товара в корзину (POST /cart/items)
3. **Настройте** CORS в FastAPI:
4. app.add\_middleware(CORSMiddleware, allow\_origins=["http://localhost:3000"])
5. ``` [restack.io](https://www.restack.io/p/fastapi-knowledge-full-stack-fastapi-postgresql-react-answer)

**День 5: Продвинутые функции магазина**

1. **Реализуйте:**
   * Категории товаров (добавьте модель Category с relation)
   * Фильтрацию через query-параметры: /products?category=electronics
   * Загрузку изображений через FastAPI FileUpload
   * Логику оформления заказа (модель Order)
2. **Добавьте** на фронтенде:
   * Страницу товара с галереей
   * Анимацию добавления в корзину (Framer Motion)
   * Хлебные крошки для навигации

**День 6: Деплой и оптимизация**

1. **Dockerize:**
2. # Бэкенд
3. FROM tiangolo/uvicorn-gunicorn-fastapi
4. COPY ./app /app
5. # Фронтенд
6. FROM node:18
7. RUN npm run build
8. **Настройте** Nginx как reverse proxy
9. **Разверните** на VPS (DigitalOcean/Vscale) или Heroku
10. **Добавьте** индексы в PostgreSQL для оптимизации:
11. CREATE INDEX idx\_products\_name ON products USING gin (name gin\_trgm\_ops);
12. ``` [help.reg.ru](https://help.reg.ru/support/servery-vps/oblachnyye-servery/ustanovka-programmnogo-obespecheniya/rukovodstvo-po-postgresql)

**День 7: Тестирование и доработки**

1. **Напишите** тесты:
   * Pytest для API (тестирование эндпоинтов)
   * Jest + Testing Library для React-компонентов
2. **Добавьте**:
   * Валидацию форм через Formik/Yup
   * Логирование ошибок (Sentry или Loguru)
   * Кэширование запросов через Redis
3. **Проверьте** нагрузку через Locust

**Полезные ресурсы:**

1. Полный пример интеграции: [[store-restack.vercel.app](https://store-restack.vercel.app/p/fastapi-knowledge-fastapi-react-project-structure)restack.io](https://store-restack.vercel.app/p/fastapi-knowledge-fastapi-react-project-structure)
2. Готовый boilerplate: [[github.com](https://github.com/tiangolo/full-stack-fastapi-postgresql)github.com](https://github.com/tiangolo/full-stack-fastapi-postgresql)
3. Шаблон аутентификации: [[auth0.com](https://auth0.com/blog/build-a-react-app-with-fastapi-and-jwt-authentication/)auth0.com/blog](https://auth0.com/blog/build-a-react-app-with-fastapi-and-jwt-authentication/)

Совет: Начните с минимальной функциональности (например, только каталог товаров), затем постепенно добавляйте фичи. Для ускорения используйте готовые компоненты из библиотек вродe Material-UI или Chakra UI.