

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP  
THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỀ THI THỰC HÀNH**

**Môn thi :** Lập trình thiết bị di động

**Lớp/Lớp học phần:** DHKTPM18B

**Ngày thi:** 15/11/2025

Thời gian làm bài: 75 phút  
(Không kể thời gian phát đề)

STT:            MSSV:                      Họ và tên SV:                                      Số máy:

**Đề 3:** Tạo ứng dụng “Expense Notes” (Expo/React Native) để quản lý các khoản chi tiêu đơn giản.

Ứng dụng lưu dữ liệu offline bằng SQLite và có chức năng import một lần danh sách chi tiêu mẫu từ API.

App có 1 màn hình danh sách chi tiêu và 1 modal để thêm/sửa.

Cấu trúc DB đề xuất:

```
expenses(  
  id INTEGER PRIMARY KEY AUTOINCREMENT,  
  title TEXT NOT NULL,  
  amount REAL NOT NULL,  
  category TEXT,  
  paid INTEGER DEFAULT 1,  
  created_at INTEGER  
)
```

Câu 1. (1đ) – Khởi tạo & cấu hình dự án

- Tạo app Expo mới, cài đặt expo-sqlite.
- Tạo repo Git, initial commit.
- Tạo file db.ts (hoặc db.js) kết nối SQLite.
- Chạy expo start đảm bảo app hoạt động bình thường.
- Commit: feat(Q1): init expo project and sqlite setup for expense notes
- Tiêu chí đạt: app chạy được, kết nối DB thành công.

Câu 2. (1đ) – Tạo bảng và seed dữ liệu mẫu

- Khi app start, tạo bảng expenses nếu chưa tồn tại.
- (Tùy chọn) Seed 2–3 khoản chi tiêu mẫu, ví dụ: “Cà phê - 30000”, “Ăn trưa - 50000”.
- Commit: feat(Q2): create expenses table and seed sample
- Tiêu chí đạt: bảng tồn tại, khởi động app không lỗi, có dữ liệu mẫu nếu lần đầu chạy.

Câu 3. (1đ) – Màn hình danh sách chi tiêu

- Tạo màn hình chính hiển thị danh sách chi tiêu bằng FlatList.

- Mỗi item hiển thị: title, amount (format có đơn vị, ví dụ: “30.000đ”), category, trạng thái paid.
- Dùng useState/useEffect để lấy dữ liệu từ SQLite.
- Empty state: hiển thị “Chưa có khoản chi tiêu nào.” nếu không có dữ liệu.
- Commit: feat(Q3): expense list screen with hooks and empty state
- Tiêu chí đạt: hiển thị danh sách hoặc empty state, không crash.

#### Câu 4. (1đ) – Thêm chi tiêu mới (Modal/Form)

- Nút “+” mở Modal thêm chi tiêu.
- Trong Modal:
  - Nhập title (bắt buộc).
  - Nhập amount (bắt buộc, số > 0).
  - Nhập category (tùy chọn).
- Khi Lưu: INSERT vào SQLite, cập nhật danh sách.
- Validate:
  - title không rỗng.
  - amount là số hợp lệ và > 0.
- Commit: feat(Q4): add expense modal with validation
- Tiêu chí đạt: thêm được khoản chi tiêu mới, dữ liệu hiển thị đúng.

#### Câu 5. (1đ) – Toggle trạng thái thanh toán (UPDATE)

- Cho phép chạm vào item để toggle paid (0 ↔ 1), ví dụ: đang nợ → đã trả.
- UI: thể hiện rõ trạng thái, có thể đổi màu hoặc icon.
- Commit: feat(Q5): toggle paid state for expense
- Tiêu chí đạt: cập nhật paid chính xác, không crash.

#### Câu 6. (1đ) – Sửa khoản chi tiêu (EDIT)

- Nhấn giữ hoặc nút “Sửa” trên từng item để mở Modal chỉnh sửa.
- Cho phép thay đổi title, amount, category.
- Lưu UPDATE vào SQLite và refresh danh sách.
- Commit: feat(Q6): edit expense via modal
- Tiêu chí đạt: chỉnh sửa đúng bản ghi, UI phản ánh dữ liệu mới.

#### Câu 7. (1đ) – Xóa khoản chi tiêu (DELETE) có xác nhận

- Thêm nút xóa hoặc swipe để xóa khoản chi tiêu.
- Hiện Alert xác nhận trước khi xóa.
- Nếu xác nhận, thực hiện DELETE và cập nhật lại danh sách.
- Commit: feat(Q7): delete expense with confirm
- Tiêu chí đạt: xóa chính xác, không xóa nhầm, không crash.

#### Câu 8. (1đ) – Tìm kiếm/Filter real-time

- Thêm TextInput Search để lọc danh sách theo title.

- (Tùy chọn) Thêm filter theo category.
- Dùng useMemo/useCallback để tối ưu việc lọc.
- Commit: feat(Q8): realtime expense search and filter
- Tiêu chí đạt: gõ là lọc; vẫn mượt với nhiều bản ghi.

Câu 9. (0.5đ) – Fetch API & Import chi tiêu mẫu

- Nút “Import từ API” trên màn hình.
- Gọi API GET lấy danh sách chi tiêu mẫu (có title, amount,...).
- Merge vào SQLite:
  - Map title → title, price/amount → amount.
  - Nếu title + amount trùng với bản ghi đã có → bỏ qua.
- Có loading và error state.
- Commit: feat(Q9): import expenses from API with merge and states
- Tiêu chí đạt: import thành công, danh sách có thêm khoản mới hợp lệ.

Câu 10. (0.5đ) – Custom hook + tổng tiền + UI/UX

- Tạo hook useExpenses đóng gói:
  - load list, insert, update, delete, search, import.
- (Tùy chọn) Tính tổng amount của tất cả chi tiêu và hiển thị dưới danh sách.
- Dùng useCallback, dọn deps useEffect.
- Nâng cấp UI: Pull to refresh, empty state đẹp, button disabled khi loading.
- Commit: feat(Q10): extract useExpenses hook and polish UI/UX
- Tiêu chí đạt: logic gọn; UI hiển thị tổng tiền, không có warning deps.

Câu 11. (1đ) – EAS build (Preview)

- Dùng EAS build tạo bản preview cho ứng dụng Expense Notes.
- Tạo file text chứa link preview, đưa vào repo.
- Commit: chore(Q11): add EAS preview link text file
- Tiêu chí đạt: có link preview hoạt động, file link tồn tại trong repo.

--- Hết ---

---

**Lưu ý:**

- Sinh viên được sử dụng mọi nguồn tài liệu để làm bài, kể cả AI (extension, github copilot, chatgpt...)
- Tuy nhiên, không được chép bài của bạn khác, hoặc từ github.
- Gởi và nhận bài đều không được chấm điểm