Transactions page challenge

Build a page showing a user's Brex Card transactions

Goal

We want to get a sense for how you:

- Understand a problem and architect a solution
- Prioritize work and make trade-offs with time constraints
- Collaborate technically with others

Agenda

Part 1: Coding

You will be coding with three frontend engineers for 45 minutes each. We'll be focusing on the following areas over the course of the interview:

- JS, CSS, and HTML
- Architecture
- Data manipulation

Part 2: Demo, Q&A

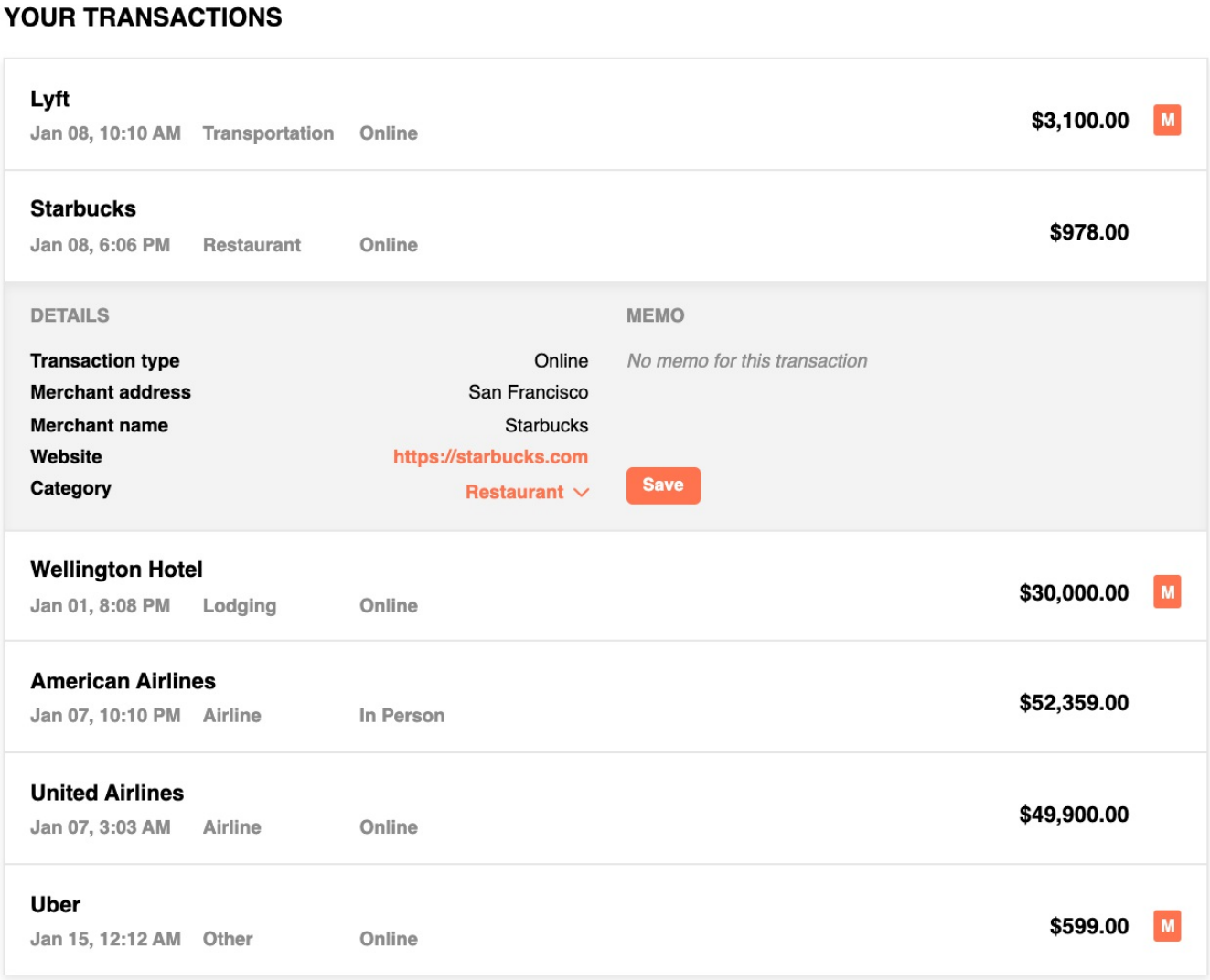
This is your chance to meet the team and ask questions. We will:

- Walk through what you've built
- Do a question and answer session about your code
- Discuss any questions you might have about Brex or the team

Spec

Users should be able to:

- View a list of Brex Card transactions received from the backend
- Determine which transactions have memos by an orange "M" icon
- Click the transaction and toggle a drawer for more information
- Change the transaction's category
- Change the transaction's memo



Data

API / data fetching

We are building a client-side application that is backed by a local API

- The API is cached once instantiated
- If you refresh the page, you should see the most up-to-date data even after the user edits a memo or category

Type definitions

- You don't have to use TypeScript to build your application, but types are provided below to describe the data you will receive by calling each method of the API
- We have a type definition file available in case you do decide to use TypeScript

```ts
type Transaction = {
  id: string;
  categoryId: string;
  accrualDate: string;
  status: string;
  amount: number;
  merchant: {
    name: string;
    website: string;
    address: string;
  };
  memo: string | null;
  cardAcceptor: {
    captureMethod: string;
  };
  card: {
    user: {
      lastName: string;
      id: string;
      firstName: string;
    };
  };
};

type Transactions = { [id: string]: Transaction };

type Category = {
  id: string;
  name: string;
};

type Categories = { [id: string]: Category };

function getTransactions(): Promise<Transactions>;

function getCategories(): Promise<Categories>;

function setTransactionCategory(
  transactionId: string,
  categoryId: string,
): Promise<Transaction>;

function setTransactionMemo(
  transactionId: string,
  memo: string,
): Promise<Transaction>;
```