You'll implement a **4-step session scheduling workflow** that allows learners to book 1-on-1 sessions with experts on the Vibly platform. This is a real-world feature that demonstrates your ability to build production-quality user interfaces.

We'll assess how you:

- Consume a GraphQL API
- Structure reusable, responsive UI
- Manage component-level & cross-step state
- Write clean, maintainable, and tested code

2. What You'll Build

You will implement the designs in this Figma File: ‣

(Dev mode Link:
https://www.figma.com/design/p2iQTRHRMEUPfv0ZXGRCL6/Interview?node-id=0-1&m=dev&t=KOV1FSDToLo2FVFq-1)

Step	Desktop Mock-up	Mobile Mock-up	Core Interaction
1. Date Selection	see asset	see asset	Choose a **date**
2. Time Selection	"	"	Choose a **time**
3. Confirmation	"	"	Review & confirm
4. Completion	"	"	Success + CTA

4 · Technical Expectations

- Framework: **Angular**.
- Language: **TypeScript**.
- Testing: Jasmine / Karma
- Styling: Any (CSS-Modules, Tailwind, Styled-Components…).
- GraphQL client: Any

5. GraphQL API Details

API KEY : "da2-ayft6maqbfeq7bytfwmkgafjty"
API URL : <https://hjddiwebzzfc5f7r7pqmqb4elq.appsync-api.us-east-1.amazonaws.com/graphql>

Header   : { "x-api-key": "da2-ayft6maqbfeq7bytfwmkgafjty" }

Data Models

```
type SessionPage {
  date: String
  time: String
  duration: Int!
  host: SessionHost!
  user: SessionUser!
  service: SessionService!
}

type SessionHost {
  photo: String
  firstName: String
  lastName: String
  profession: String
}

type SessionUser {
  email: String
  firstName: String
  lastName: String
  timeZone: String
}

type SessionService {
```

```
  title: String!
}

type DateAvailability {
 date: String!
 times: [TimeRange!]
}

type TimeRange {
 start: AWSTime
 end: AWSTime
}

type Query {
 getSessionPage(id: ID!): SessionPage!
 getAvailableDateTimes(userId: ID!): [DateAvailability!]!
}
```

Query 1 · Get Session Page

```
query GetSessionPage( {
 getSessionPage(
                id: "..." // You can use an UUID here
        ) {
   date
   time
   duration
   host { firstName lastName photo profession }
   user { firstName lastName email timeZone }
   service { title }
 }
}
```

Example Response:

```json
{
  "data": {
    "getSessionPage": {
      "date": "2025-08-08",
      "time": "10:00:00",
      "duration": 30,
      "host": {
        "firstName": "Jane",
        "lastName": "Doe",
        "photo": "<https://example.com/jane.jpg>",
        "profession": "Leadership Coach"
      },
      "user": {
        "firstName": "John",
        "lastName": "Smith",
        "email": "john.smith@vibly.io",
        "timeZone": "Europe/Athens"
      },
      "service": {
        "title": "Coaching Session"
      }
    }
  }
}
```

Query 2 · Get Available Date Times

```
query GetAvailableDateTimes {
 getAvailableDateTimes(
                userId: "..." // You can use an UUID here
        ) {
   date
   times { start end }
 }
}
```

Example Response:

```json
{
  "data": {
    "getAvailableDateTimes": [
      {
        "date": "2025-08-04",
        "times": [
          { "start": "10:00:00", "end": "12:00:00" },
          { "start": "13:00:00", "end": "16:00:00" }
        ]
      },
      {
        "date": "2025-08-05",
        "times": [
          { "start": "09:00:00", "end": "12:00:00" },
          { "start": "14:00:00", "end": "17:00:00" }
        ]
      },
      {
        "date": "2025-08-06",
        "times": [
          { "start": "10:00:00", "end": "12:00:00" },
          { "start": "13:00:00", "end": "16:00:00" }
        ]
      }
      // ...more dates for the month
    ]
  }
}
```

Data Source Guide

Step 1: Date Selection

Left Column (Session Details)

- **Host avatar, name, profession:** → Load from getSessionPage(id) → host.photo, host.firstName, host.lastName, host.profession
- **Session type ("1-1 session"):** → **Static** (always the same)
- **Service title:** → Load from getSessionPage(id) → service.title
- **Session duration:** → Load from getSessionPage(id) → duration
- **Meeting platform ("Google meets"):** → **Static** (always the same)

Center Column (Calendar)

- **Available dates:** → Load from getAvailableDateTimes(userId) Use the date field from the response to highlight/select available days.
- **Disable past dates & unavailable days**

Step 2: Time Selection

Schedule session

Jane Doe
Leadership Coach

1-1 session
Coaching Session

🕐 30min

Google meets

Select date:
Select a date to see the available time slots on that day

◀ August 2025 ▶

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	**8**	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Time Zone
Eastern European Time (11:04)

Thursday, 8 August 2025

Select time:

- 8:00am
- 8:30am
- 9:00am
- 10:00am
- 10:15am
- 10:30am
- 10:45am
- 11:30am
- 1:30pm
- 2:30pm
- 2:45pm
- 3:00pm

Left Column (Session Details)

- **Same as Step 1:** → Use data from getSessionPage(id)

Center Column (Calendar)

- **Same as Step 1:** → Use data from getAvailableDateTimes(…)

Right Column

- **Available time slots for selected date:** → Use the times array for the selected date from getAvailableDateTimes(…)

Step 3: Confirmation

Schedule session

POWERED BY Vibly

Jane Doe
Leadership Coach

1-1 session
Coaching Session

🕐 30min

Google meets

Confirm booking

Thursday, 8 August 2025
8:00am - 8:30am (EET)
Eastern European Time (11:04)

John Smith
john.smith@vibly.io

We will send you a confirmation email with the session details and meeting link.

Back

Confirm

Left Column (Session Details)

- **Same as Step 1:** → Use data from getSessionPage(id)

Main Section

- **Selected date and time:** → From user's selection in previous steps (date from calendar, time from time slot)
- **User details (name, email):** → Load from getSessionPage(id) → user.email, user.firstName, user.lastName
- **Timezone:** → Load from getSessionPage(id) → user.timeZone

Step 4: Completion

Booking Successful

Jane Doe

Leadership coach

Take your growth to the next level

Join Vibly

Main Section

- **Success checkmark, "Booking Successful", tagline:** → **Static** (UI only)
- **Host avatar and name:** → Use data from getSessionPage(id) → host.photo, host.firstName, host.lastName
- **"Join Vibly" CTA:** → **Static** link to https://vibly.io

Submission

1. Push code to **your own GitHub repo**
2. Ensure README.md (setup) and NOTES.md (decisions, trade-offs) are present.
3. Email the repository link and any extra notes to jad@vibly.io

We **encourage** you to use AI-powered tools to help you work faster and smarter—just as you would in a real-world engineering environment. We believe great engineers use the best tools available, and we want to see how you leverage AI to boost your productivity

Good luck — we're excited to see what you build!