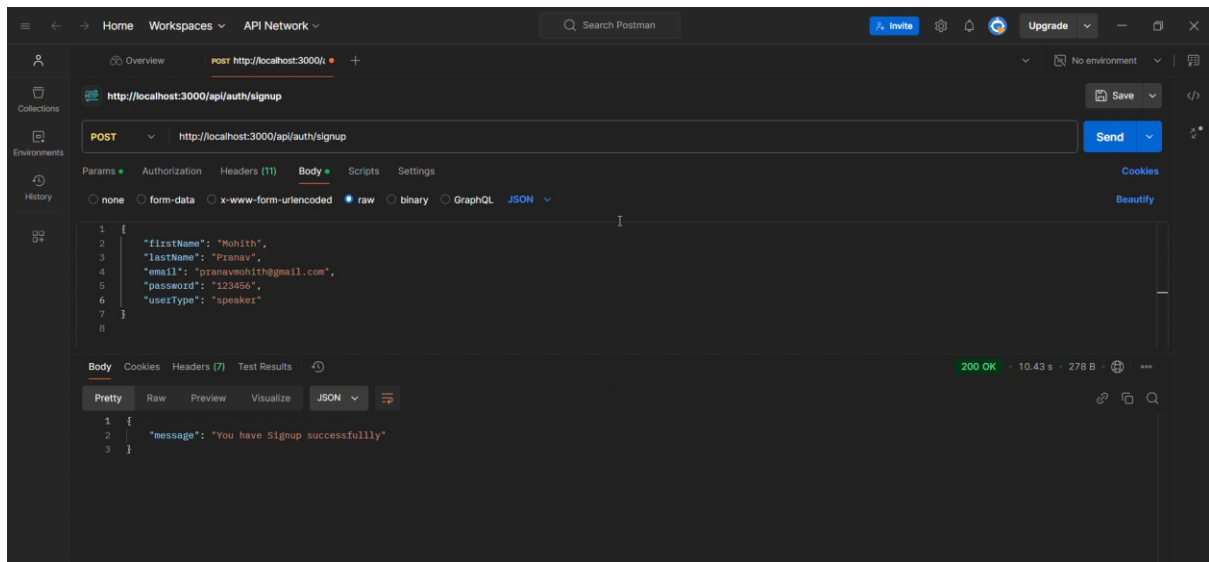


API Documentation for Speaker Session Booking

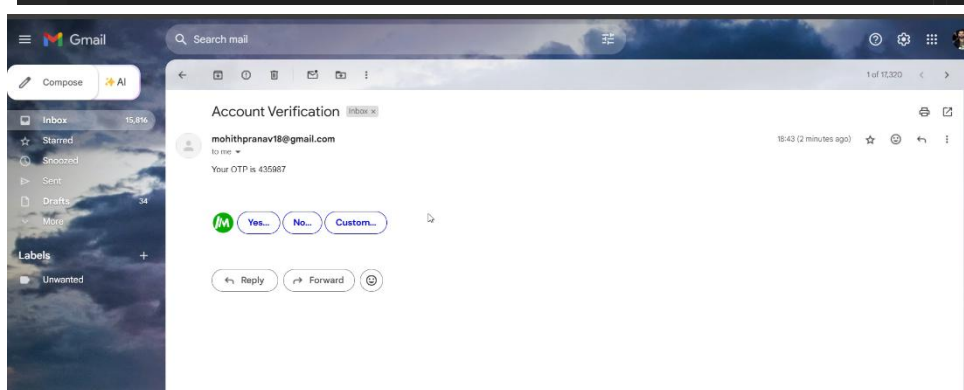
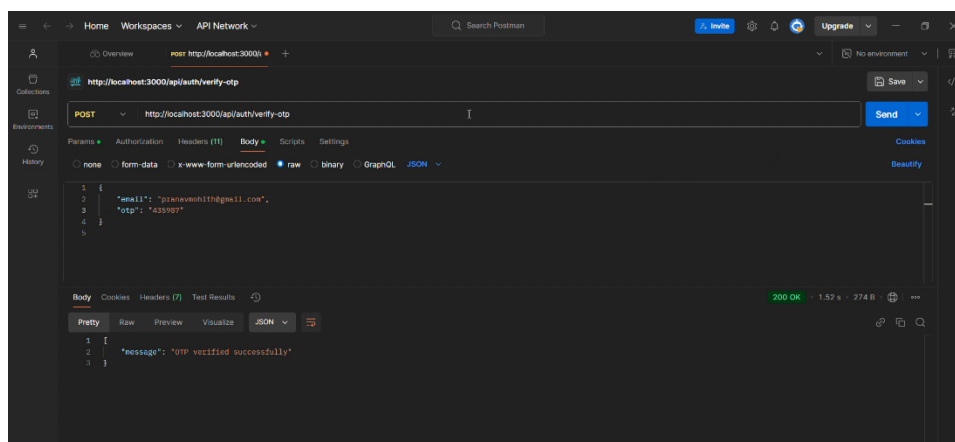
1. Signup

- **Endpoint:** POST /api/auth/signup
- **Description:** Allows a user or speaker to create an account.
- **Body Parameter as give below**



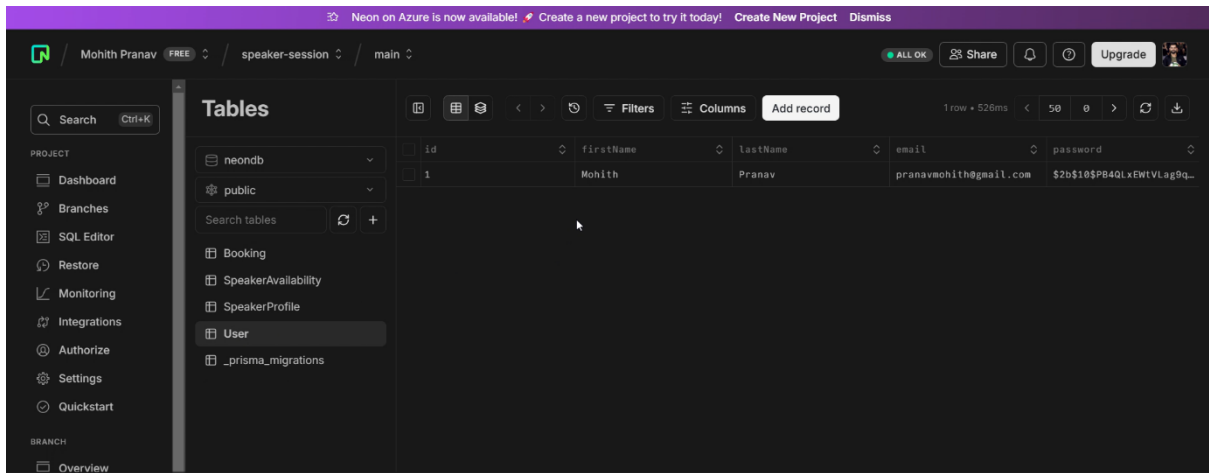
2. Verify OTP

- **Endpoint:** POST /api/auth/verify-otp
- **Description:** Verifies the OTP sent to the user's email.
- **Body Parameter as give below**



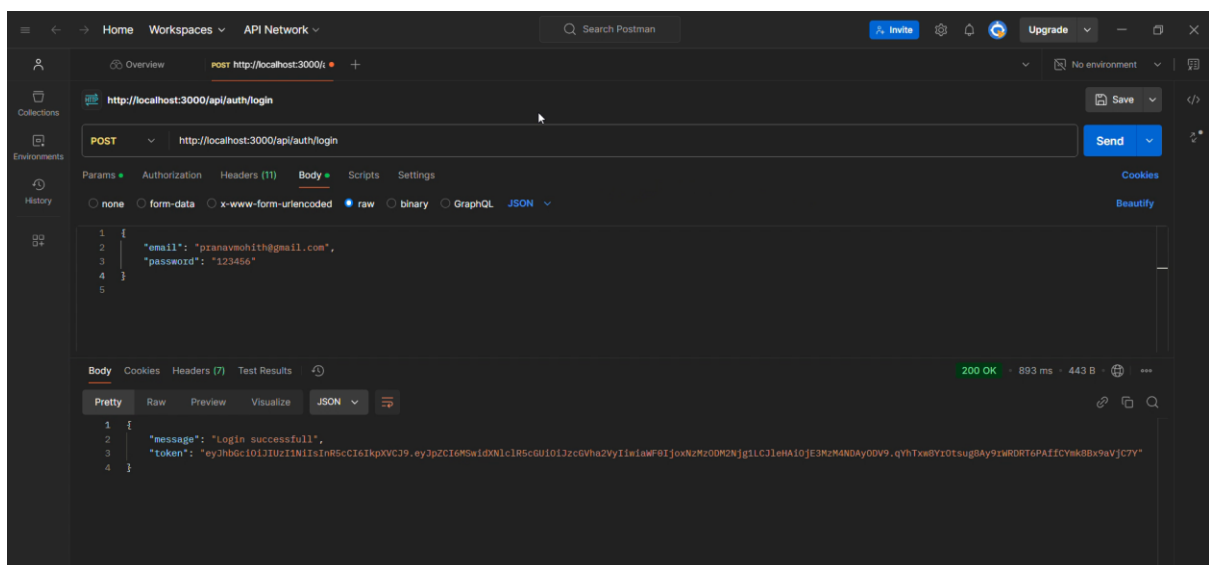
API Documentation for Speaker Session Booking

- **Database:** Created speaker to the database



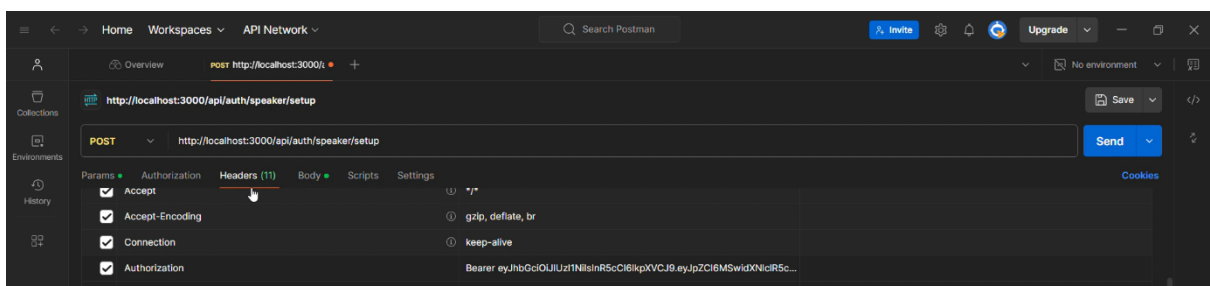
3. Login

- **Endpoint:** POST /api/auth /login
- **Description:** Allows a user or speaker to login and JWT generated



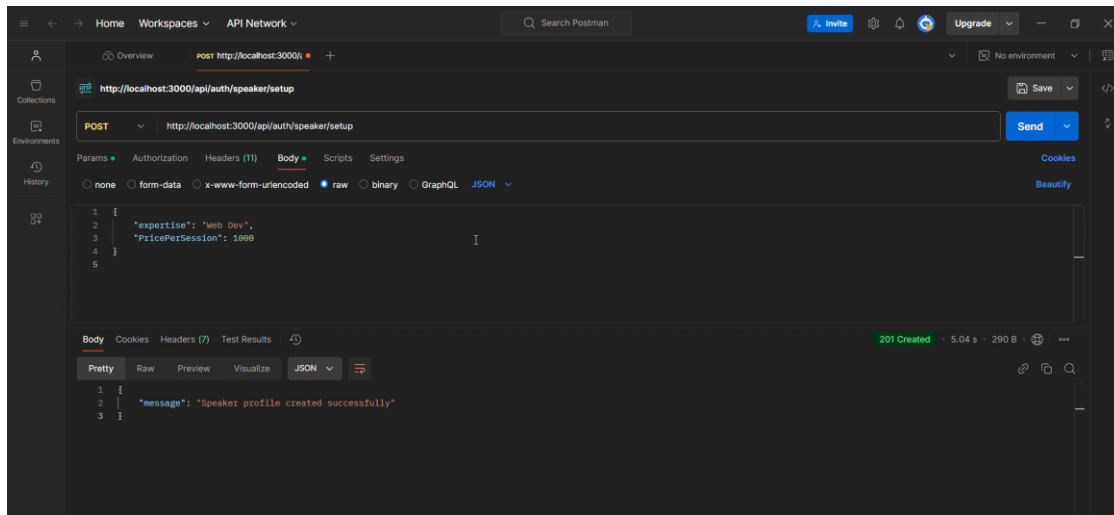
4. Setup Speaker Profile

- **Endpoint:** POST /api/auth /speaker/setup
- **Description:** Allows a speaker to set up their profile needs the headers
- **Header as Authorization:** Bearer <token> for speaker authentication

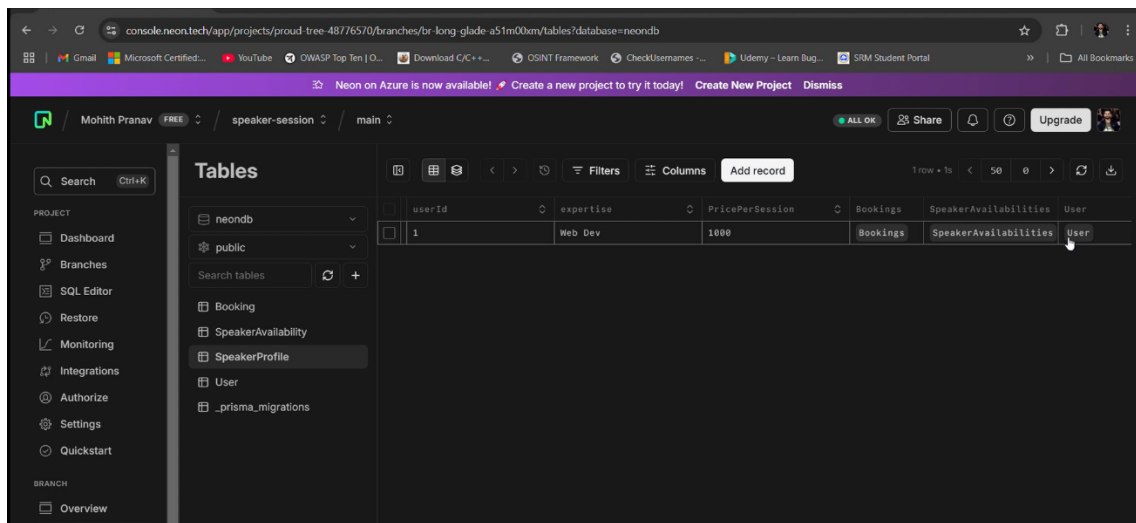


API Documentation for Speaker Session Booking

- **Body Parameter as give below**



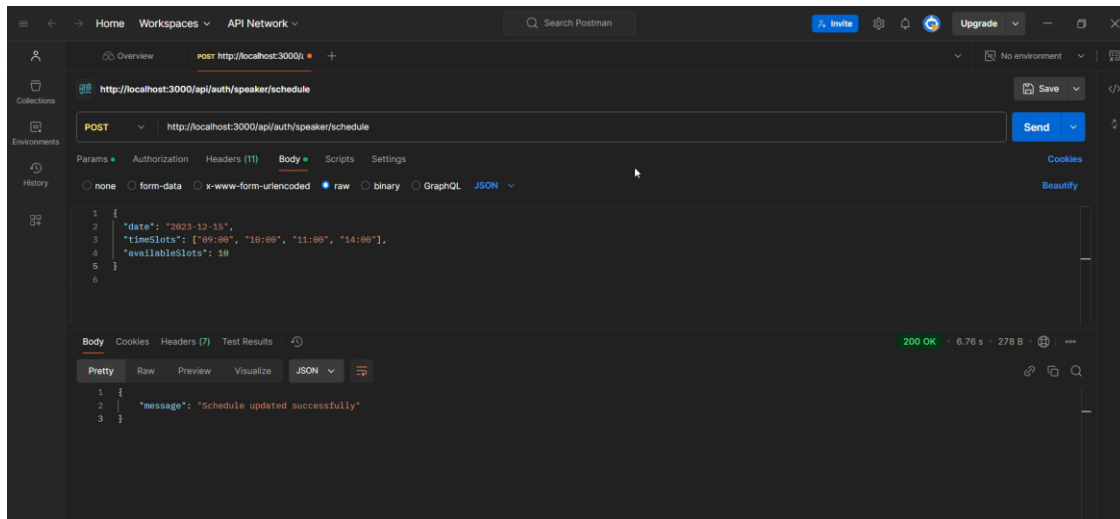
- **Added speaker profile in database:**



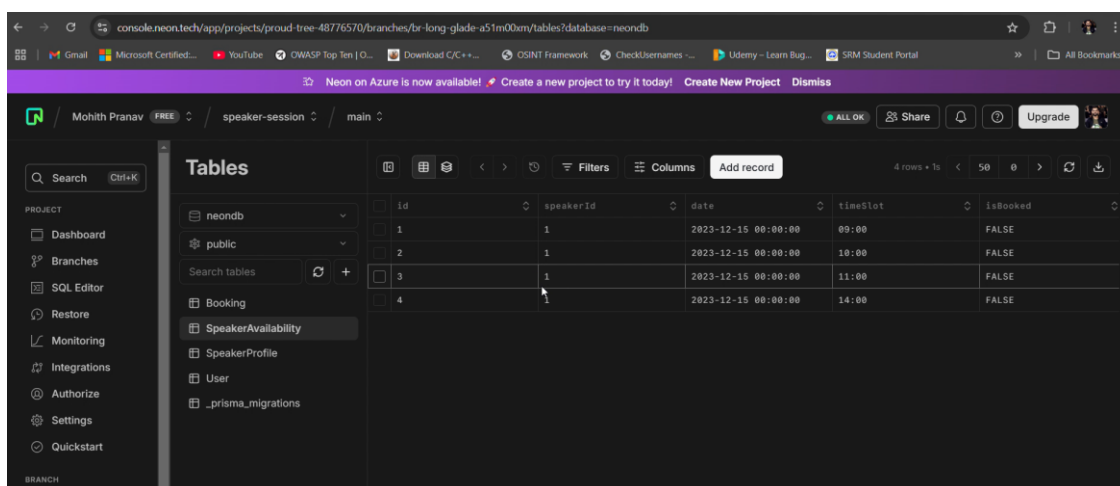
API Documentation for Speaker Session Booking

5. Set Schedule

- **Endpoint:** POST /api/auth /schedule
- **Description:** Allows a speaker to set their availability for bookings.
- **Headers:**
 - **Authorization:** Bearer <token> for speaker authentication
- **Body Parameter** as give below



- Added speaker Availability in database:

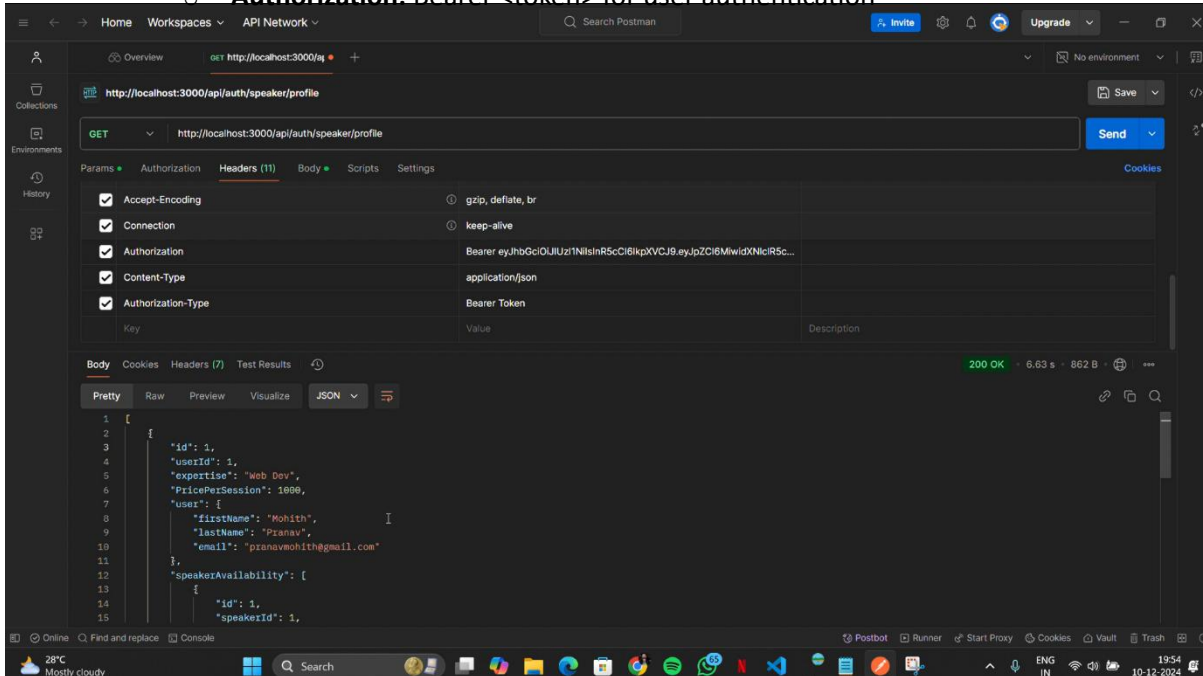


6. Get Speakers

API Documentation for Speaker Session Booking

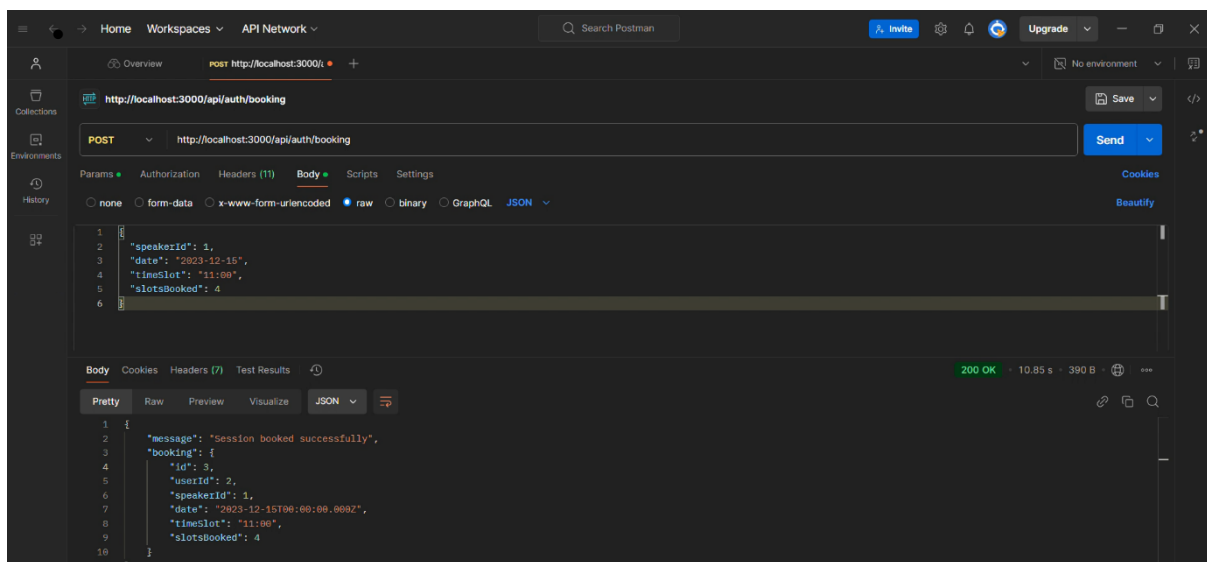
- **Endpoint:** GET /api/auth /speaker/profile
- **Description:** Fetch a list of speakers available for booking.
- **Headers:**

- **Authorization:** Bearer <token> for user authentication



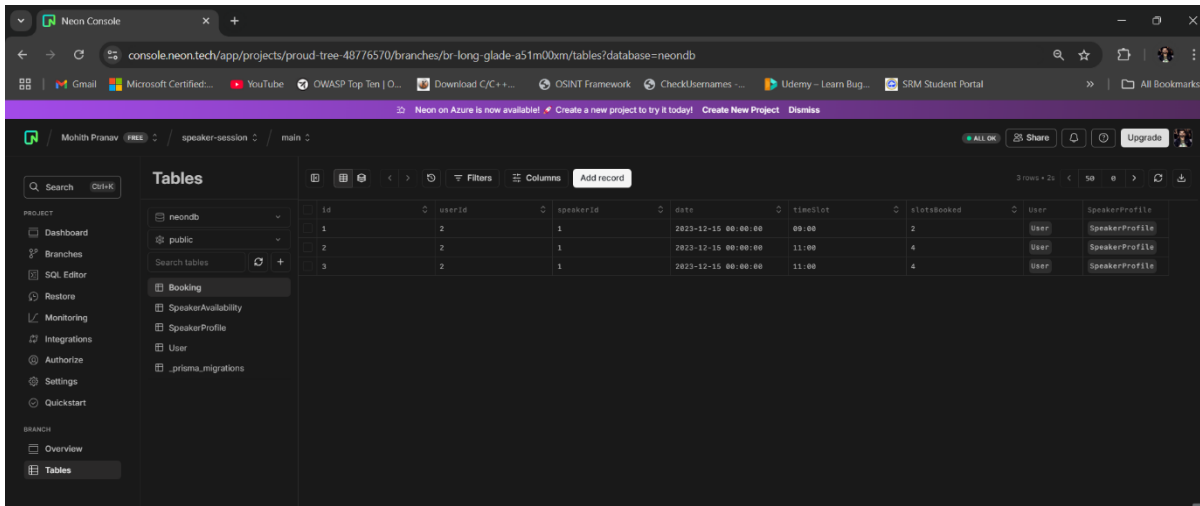
7. Book a Session

- **Endpoint:** POST /api/auth /booking
- **Description:** Allows a user to book a session with a speaker.
- **Headers:**
 - **Authorization:** Bearer <token>
- **Body Parameter as give below**



API Documentation for Speaker Session Booking

- Booking of user is added in database

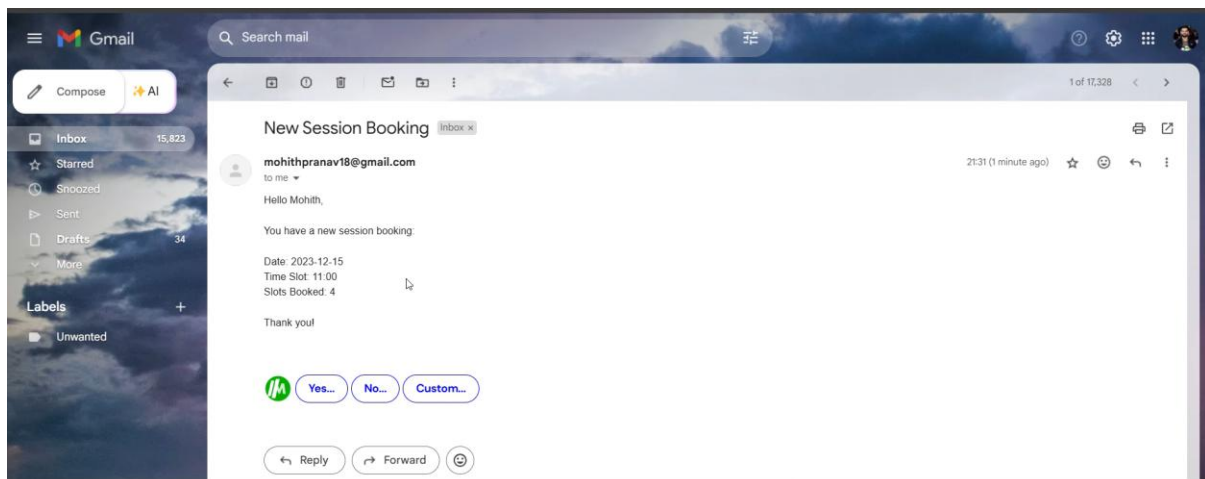


The screenshot shows the Neon Console interface. The 'Tables' section on the left lists several tables: 'neondb', 'public', 'Booking', 'SpeakerAvailability', 'SpeakerProfile', 'User', and '_prisma_migrations'. The 'Booking' table is selected, and its data is displayed in a table with the following columns: 'id', 'userId', 'speakerId', 'date', 'timeSlot', 'slotsBooked', 'User', and 'SpeakerProfile'.

id	userId	speakerId	date	timeSlot	slotsBooked	User	SpeakerProfile
1	2	1	2023-12-15 00:00:00	00:00	2	User	SpeakerProfile
2	2	1	2023-12-15 00:00:00	11:00	4	User	SpeakerProfile
3	2	1	2023-12-15 00:00:00	11:00	4	User	SpeakerProfile

8. Email Notification on Booking

- **Trigger:** Upon successful booking, sends the conformation mail to both user and speaker



API Documentation for Speaker Session Booking

