Backend Basics: Save "Need Help" Posts

Goal:
- Save Need Help posts in Supabase.
- Return each user only their own posts.

1) Create table in Supabase

Run this SQL:

```sql
create extension if not exists "uuid-ossp";

create table if not exists public.need_help_posts (
  id uuid primary key default uuid_generate_v4(),
  user_id uuid not null references public.app_users(id),
  categories text[] not null,
  details text not null,
  city text,
  created_at timestamptz not null default now()
);
```

Also add this table to `backend/src/main/resources/db/create_tables.sql`.

2) Add backend files (same project pattern)

Create:
- `entity/NeedHelpPost.java`
- `repository/NeedHelpPostRepository.java`
- `service/NeedHelpPostService.java`
- `controller/NeedHelpPostController.java`
- DTOs for request/response

Repository should include:
- `findByUserIdOrderByCreatedAtDesc(UUID userId)`

3) Add 2 endpoints

- `POST /api/need-help/posts` (authenticated)
- `GET /api/need-help/posts/mine` (authenticated)

In controller:
- Read logged-in user from `AuthPrincipal` (same style as `InviteController`).
- Use `userId` from auth, not from request body.

4) Expected payload and response

Request example:

```json
{
  "categories": ["alimentos"],
  "details": "Necesito apoyo con alimentos esta semana",
  "city": "Salt Lake City"
}
```

Create response example:

```json
{
```

~~~~markdown
  "id": "uuid",
  "categories": ["alimentos"],
  "details": "Necesito apoyo con alimentos esta semana",
  "city": "Salt Lake City",
  "createdAt": "2026-02-14T12:00:00.000Z"
}
```

Error format:

```json
{ "error": "Unauthorized" }
```

## 5) Frontend switch when backend is ready

Set:

```bash
NEXT_PUBLIC_NEEDHELP_MODE=api
```

Frontend is already wired and will call:
- `POST /api/need-help/posts`
- `GET /api/need-help/posts/mine`
~~~~