

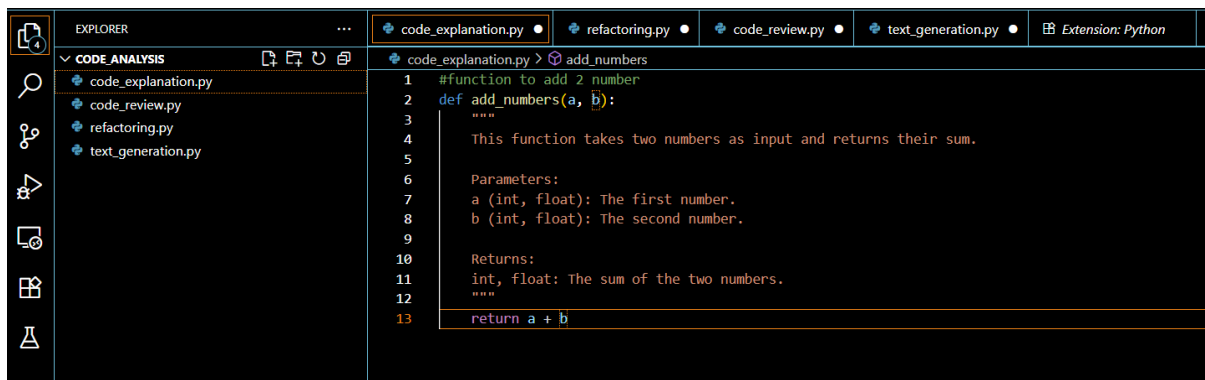
Assignment-3

1 Question

By using github copilot we can do so many tasks which may take less time to code a huge problem. By just giving the problem in comments it will be generating code using ai.

Here are some of them,

1) Code Explanation

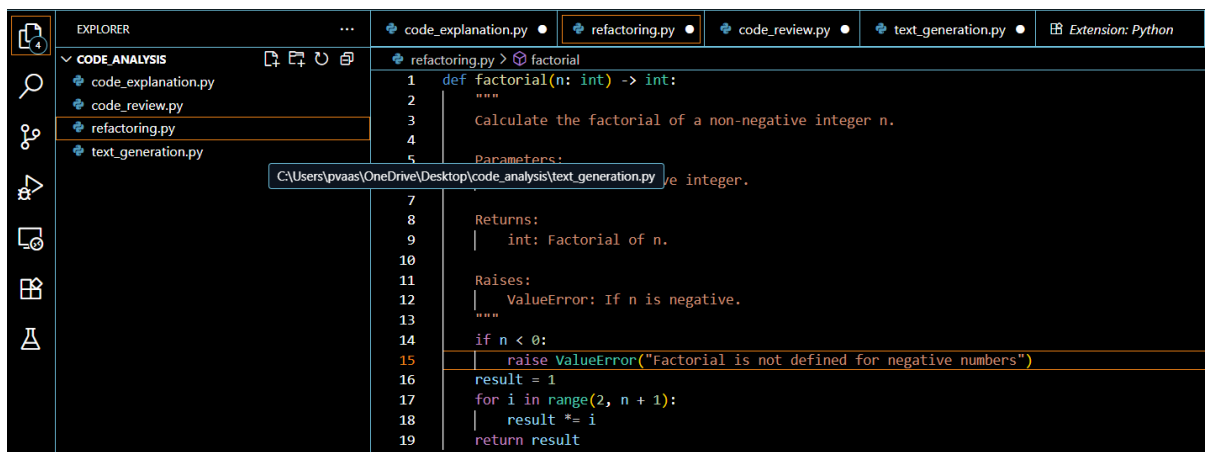


The screenshot shows the VS Code interface with the Explorer sidebar on the left displaying a folder named 'CODE ANALYSIS' containing files: code_explanation.py, code_review.py, refactoring.py, and text_generation.py. The main editor window has tabs for code_explanation.py, refactoring.py, code_review.py, and text_generation.py, with the Python extension installed. The code_explanation.py tab is active, showing a Python function named 'add_numbers' with a docstring explaining its purpose and parameters. The function takes two numbers as input and returns their sum.

```
1 #function to add 2 number
2 def add_numbers(a, b):
3     """
4     This function takes two numbers as input and returns their sum.
5
6     Parameters:
7     a (int, float): The first number.
8     b (int, float): The second number.
9
10    Returns:
11    int, float: The sum of the two numbers.
12    """
13    return a + b
```

By using co-pilot we can get explanation which helps us to understand what are the right and wrongs in code. If there are any problem in code it will be correcting them and explains us what is wrong in it.

2)Code Refactoring

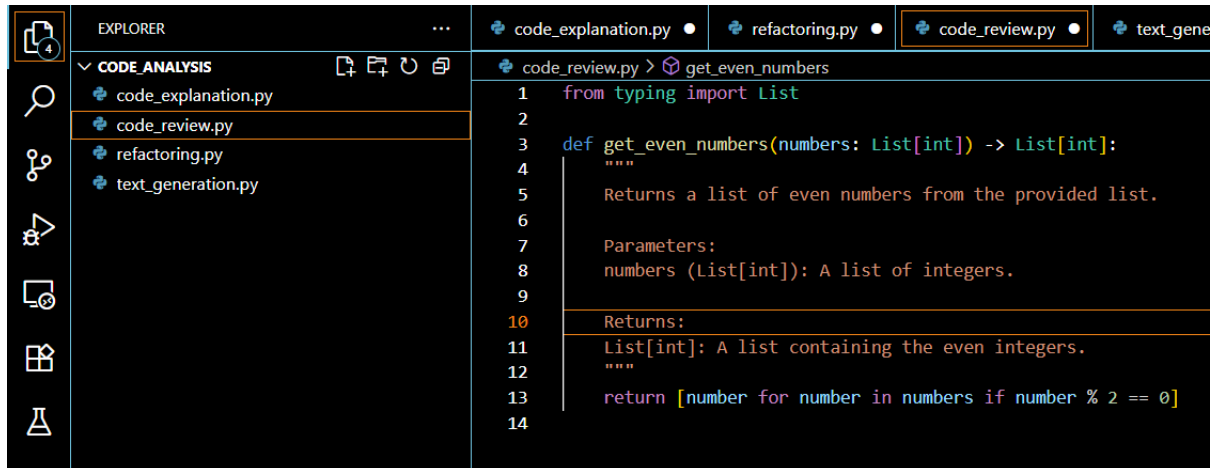


The screenshot shows the VS Code interface with the Explorer sidebar on the left displaying a folder named 'CODE ANALYSIS' containing files: code_explanation.py, code_review.py, refactoring.py, and text_generation.py. The main editor window has tabs for code_explanation.py, refactoring.py, code_review.py, and text_generation.py, with the Python extension installed. The refactoring.py tab is active, showing a Python function named 'factorial' with a docstring explaining its purpose and parameters. The function calculates the factorial of a non-negative integer n. A tooltip is visible over the file path 'C:\Users\pvaas\OneDrive\Desktop\code_analysis\text_generation.py'.

```
1 def factorial(n: int) -> int:
2     """
3     Calculate the factorial of a non-negative integer n.
4
5     Parameters:
6     n (int): A non-negative integer.
7
8     Returns:
9     int: Factorial of n.
10
11    Raises:
12    ValueError: If n is negative.
13    """
14    if n < 0:
15        raise ValueError("Factorial is not defined for negative numbers")
16    result = 1
17    for i in range(2, n + 1):
18        result *= i
19    return result
```

We can get refactoring option which helps us to make code effective and help us to understand.

3)Code Review



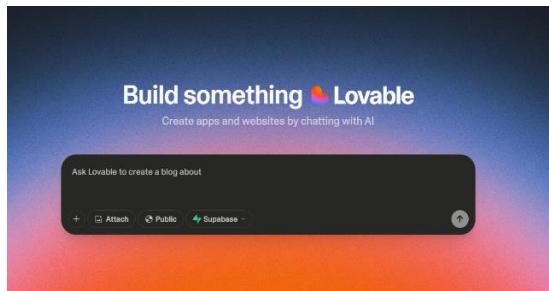
```
1 from typing import List
2
3 def get_even_numbers(numbers: List[int]) -> List[int]:
4     """
5     Returns a list of even numbers from the provided list.
6
7     Parameters:
8     numbers (List[int]): A list of integers.
9
10    Returns:
11    List[int]: A list containing the even integers.
12    """
13    return [number for number in numbers if number % 2 == 0]
14
```

It will review the code and tell the mistakes and easy way to solve it in one or few steps

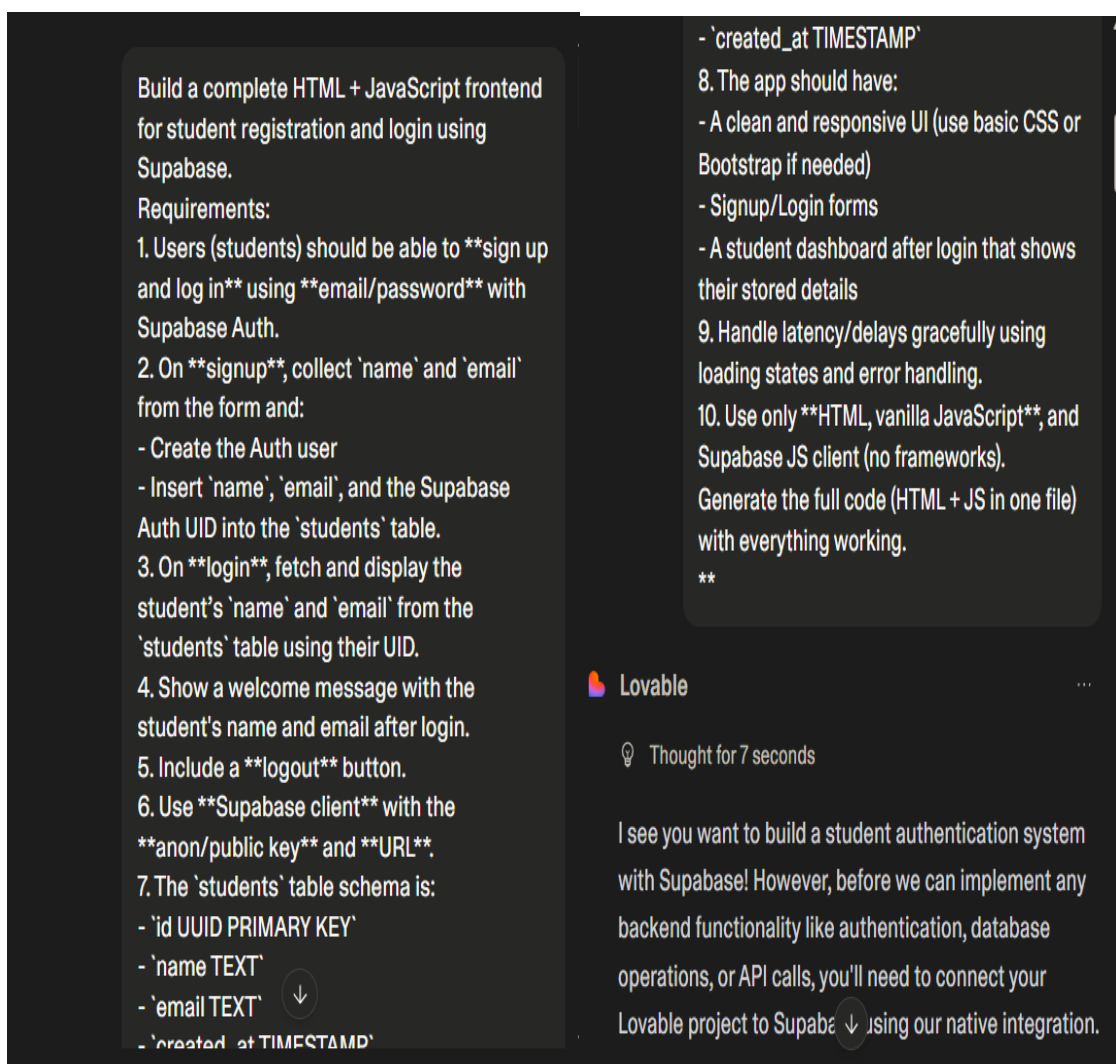
Ex:-

In above code I have given list of numbers and ask it to find even numbers among them, It uses **list comprehensions** to make it simple and to complete in few steps

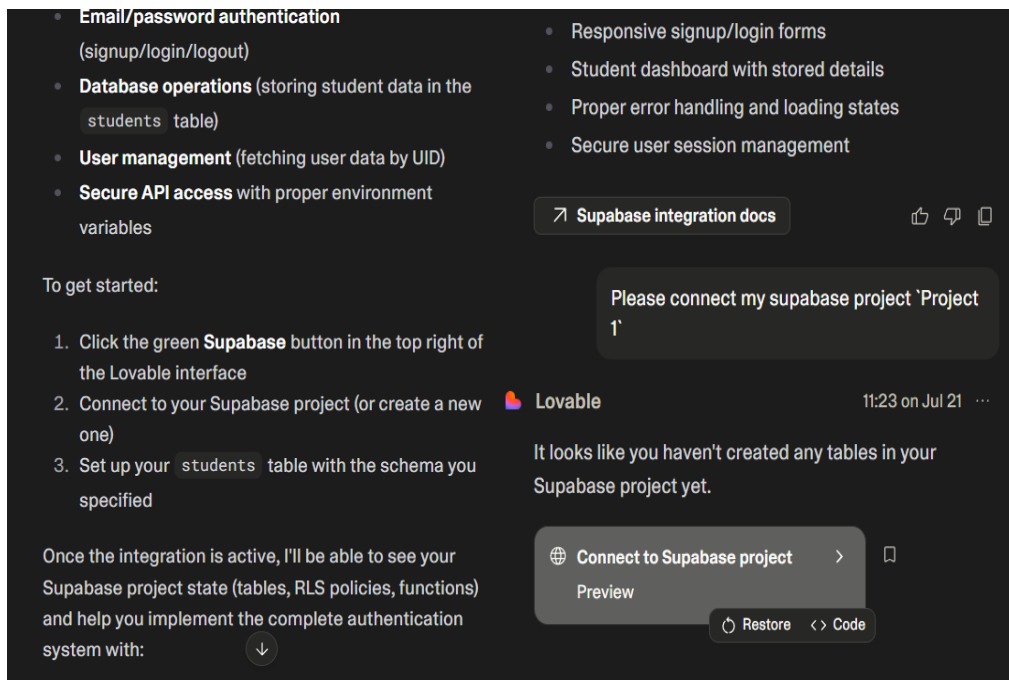
2 Question



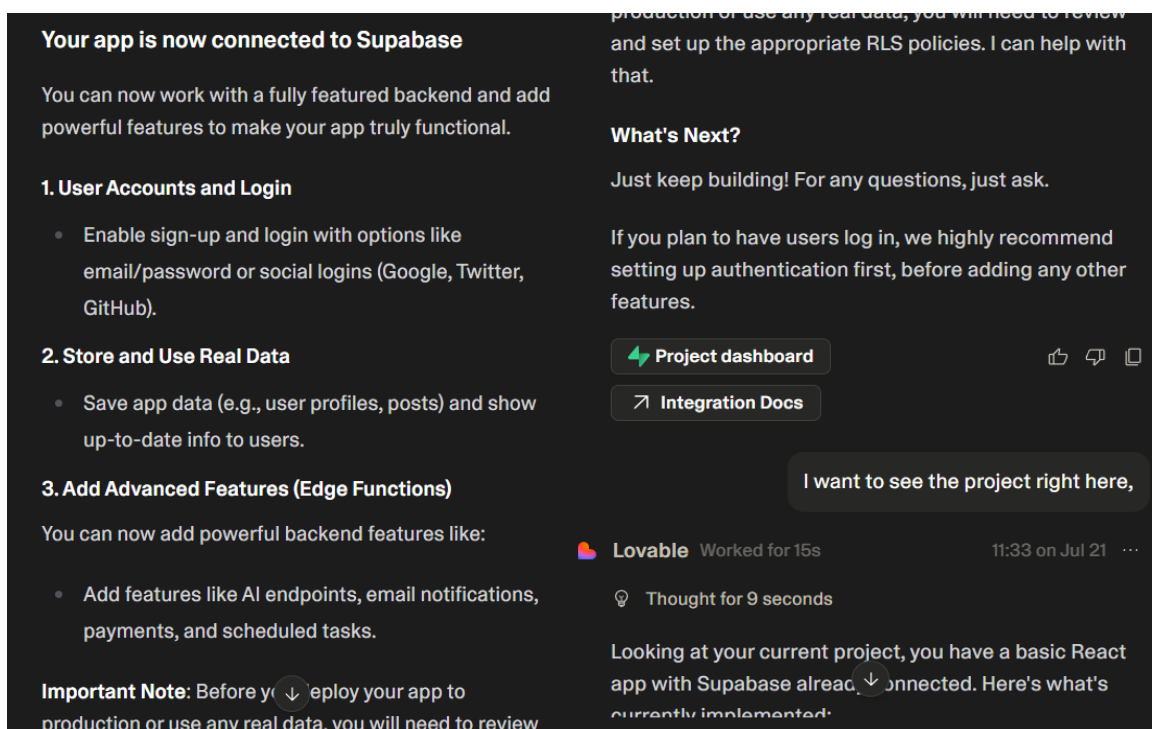
This is the interface of Lovable after login into it. Then we will integrate with Supabase project. Then the upcoming interface will be coming into place.



After giving this prompt we will be getting linked to Supabase project.



After some time it will ask us to integrate our project to supabase, so we will give the prompt to do the work.



After integrating with our supabase project it will be setting all components in our project and asking us to "do you need any help?". We will be asking to display the project right here

Current State:

- Basic React app with routing setup
- Supabase client configured and ready to use
- A blank landing page showing "Welcome to Your Blank App"
- No authentication or student registration functionality yet

What you can see:

- The app is running on the Index page (/)
- Supabase integration is ready but no database tables or auth flows are implemented
- The project uses React, TypeScript, Tailwind CSS, and has UI components ready

Next steps for your student auth system: You'll need to decide if you want me to:

1. Create the database schema (students table)
2. Implement the signup/login pages

system for you. Let me start by creating the database schema and then build the authentication pages.

```
-- Create a profiles table to store student information
CREATE TABLE public.profiles (
  id UUID NOT NULL DEFAULT gen_random_uuid()
  PRIMARY KEY,
  user_id UUID NOT NULL REFERENCES
auth.users(id) ON DELETE CASCADE,
  email TEXT NOT NULL,
  first_name TEXT,
  last_name TEXT,
  student_id TEXT UNIQUE,
  phone TEXT,
  date_of_birth DATE,
  address TEXT,
  created_at TIMESTAMPT WITH TIME ZONE NOT NULL
  DEFAULT now(),
  updated_at TIMESTAMPT WITH TIME ZONE NOT NULL
  DEFAULT now(),
  UNIQUE(user_id)
);

-- Enable Row Level Security
ALTER TABLE public.profiles ENABLE ROW LEVEL
SECURITY;
```

```
-- Create policies for user access
CREATE POLICY "Users can view their own profile"
ON public.profiles
FOR SELECT
USING (auth.uid() = user_id);

CREATE POLICY "Users can create their own profile"
ON public.profiles
FOR INSERT
WITH CHECK (auth.uid() = user_id);

CREATE POLICY "Users can update their own profile"
ON public.profiles
FOR UPDATE
USING (auth.uid() = user_id);

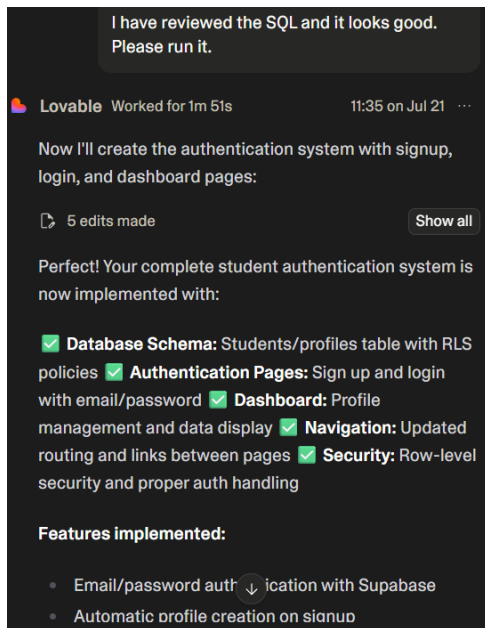
-- Create function to update timestamps
CREATE OR REPLACE FUNCTION
public.update_updated_at_column()
RETURNS TRIGGER AS $$
BEGIN
  NEW.updated_at = now();
  RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
updates
CREATE TRIGGER update_profiles_updated_at
BEFORE UPDATE ON public.profiles
FOR EACH ROW
EXECUTE FUNCTION
public.update_updated_at_column();

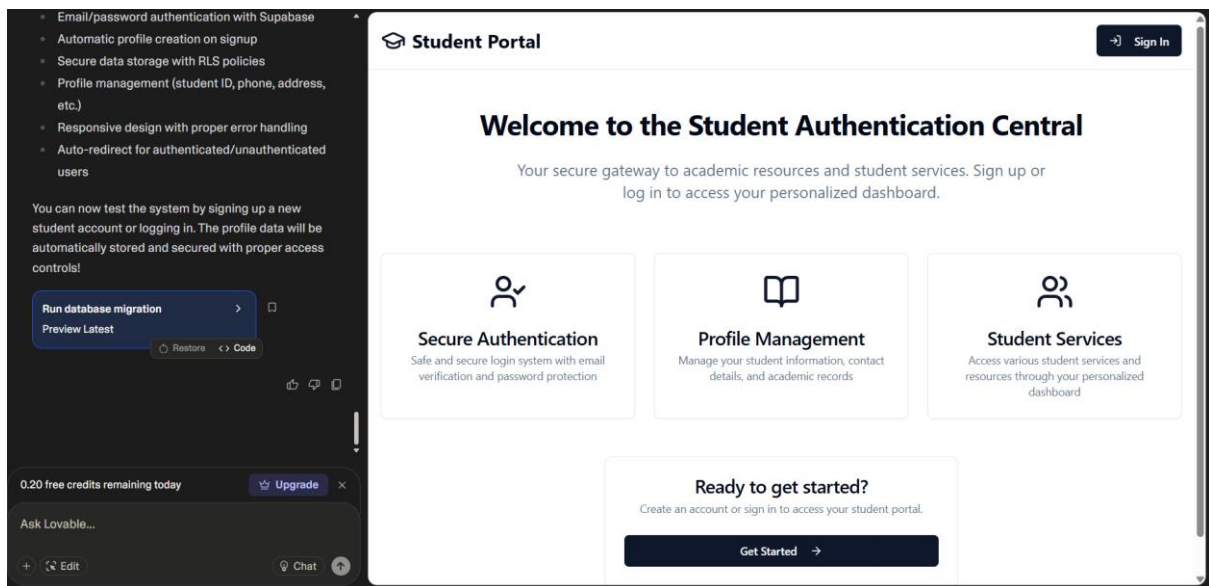
-- Create function to handle new user registration
CREATE OR REPLACE FUNCTION
public.handle_new_user()
RETURNS TRIGGER AS $$
BEGIN
  INSERT INTO public.profiles (user_id, email,
first_name, last_name)
VALUES (
  NEW.id,
  NEW.email,
  NEW.raw_user_meta_data --> 'first_name',
  NEW.raw_user_meta_data --> 'last_name'
);
  RETURN NEW;
END;
$$ LANGUAGE plpgsql SECURITY DEFINER;

-- Create trigger to automatically create profile when user signs up
CREATE TRIGGER on_auth_user_created
AFTER INSERT ON auth.users
FOR EACH ROW
EXECUTE FUNCTION public.handle_new_user();
```

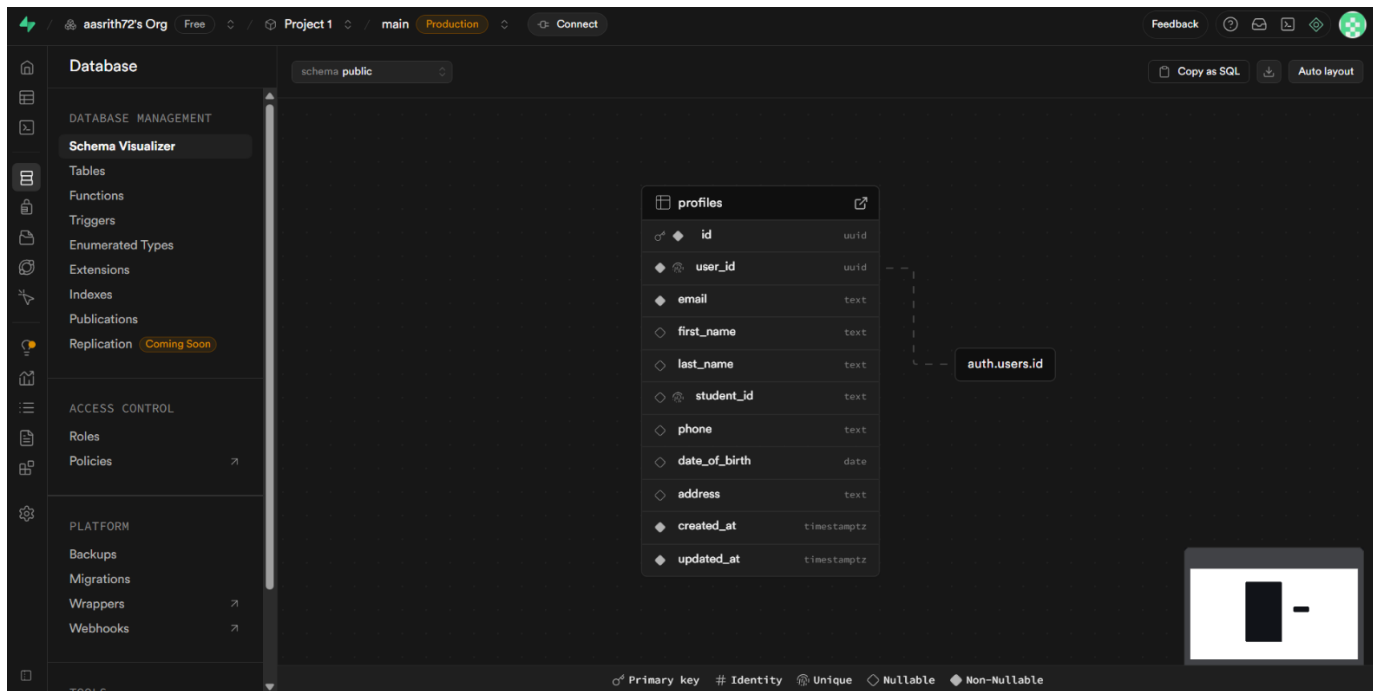
After running the prompt we given it will be given the sql code for our database in supabase project which we integrated with lovable. After it is asking that do I need to run it here and we will be saying yes please run here.



After some time the project will be appearing as we shown below.

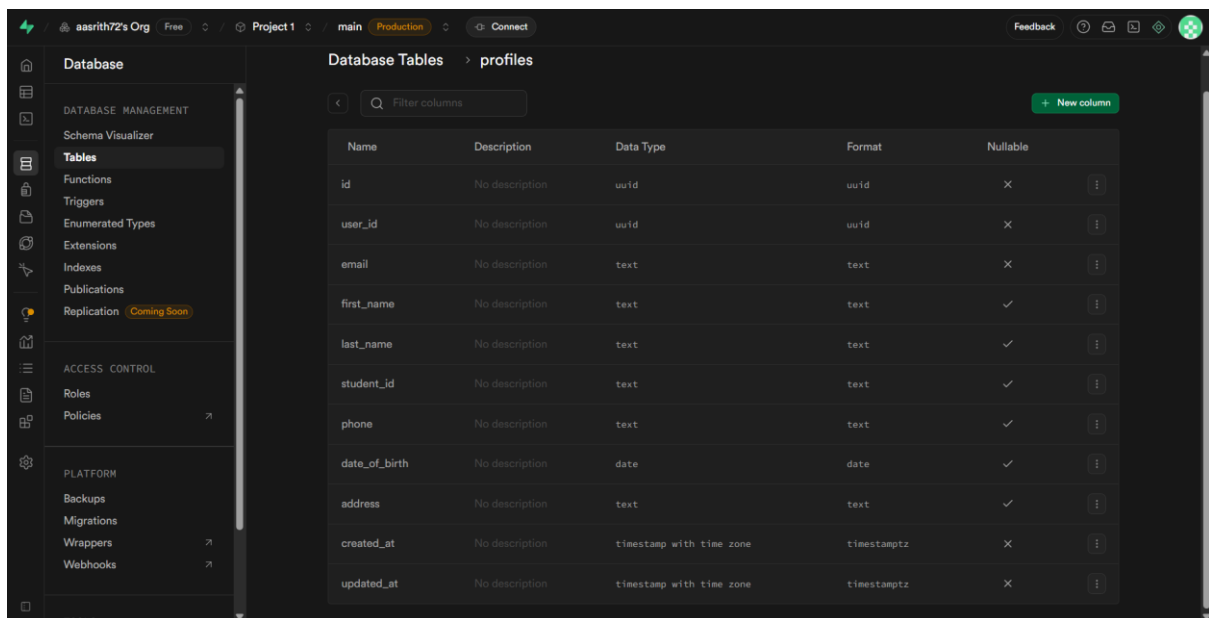


Finally, After running the prompt the project is done and we can see the supabase project code also.



Supabase interface:-

In database section, we will be seeing our components of our project and we can update. There are so many sections at left side where we will be seeing tables and some etc options for our project.



Here, we can see the table which is created by lovable when we integrated with supabase.

***** THE END *****