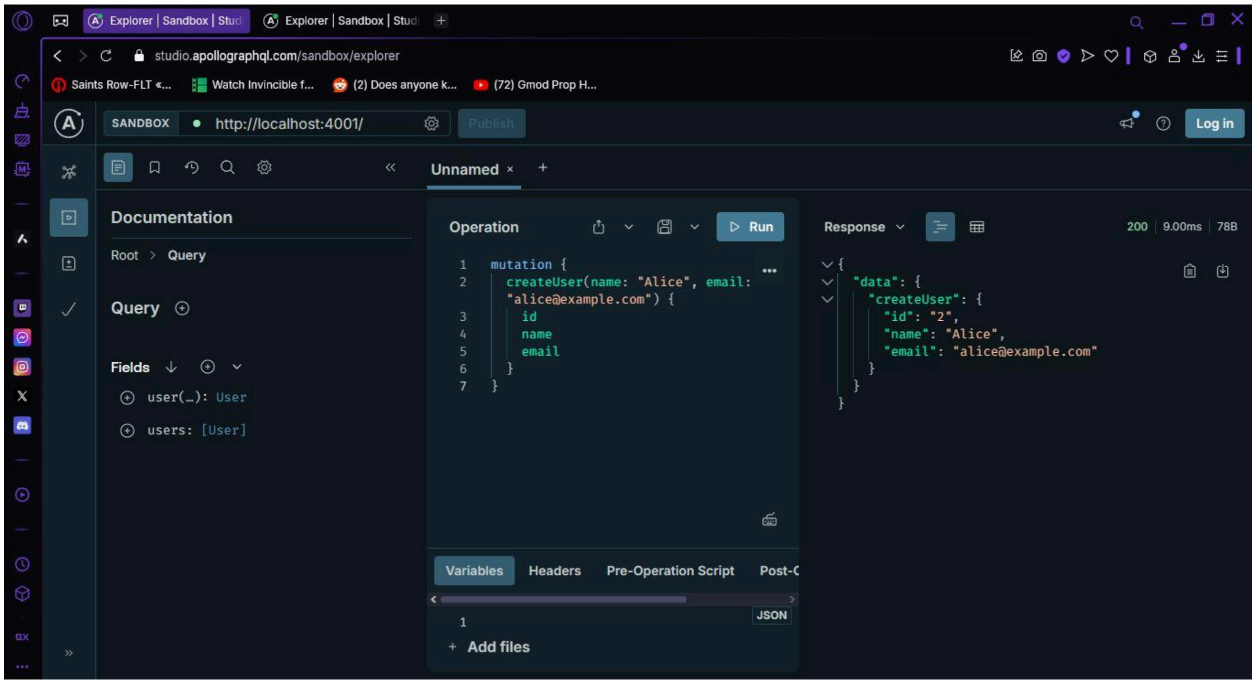
Users Service - Create, Read, Update, and Delete Records Verification:

Create Record:

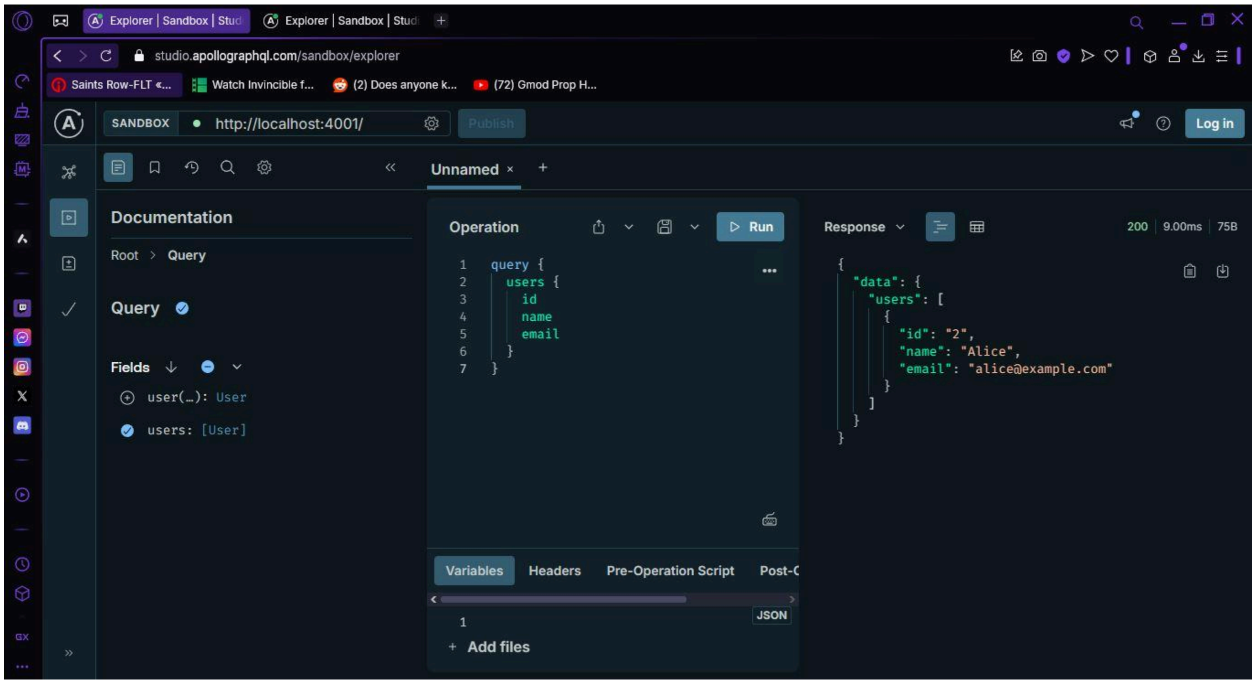


```
mutation {
  createUser(name: "Alice", email:
  "alice@example.com") {
    id
    name
    email
  }
}
```

Response 200 9.00ms 78B

```
{
  "data": {
    "createUser": {
      "id": "2",
      "name": "Alice",
      "email": "alice@example.com"
    }
  }
}
```

Read Record:

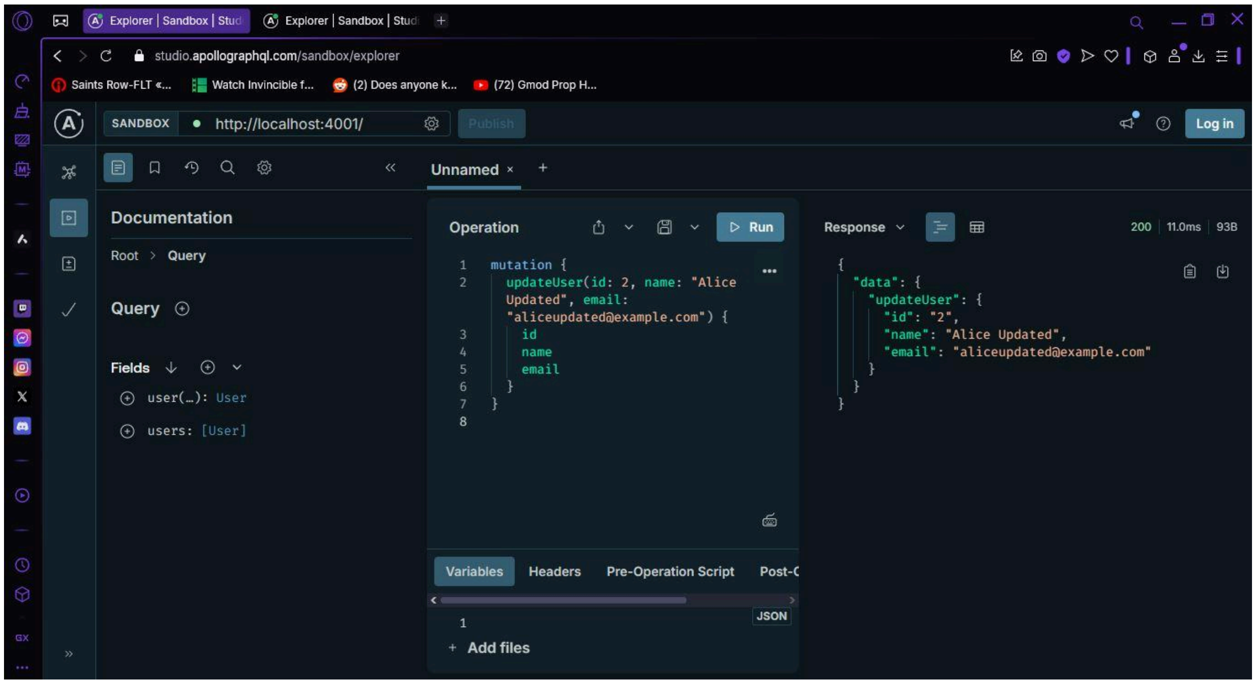


```
query {
  users {
    id
    name
    email
  }
}
```

Response 200 9.00ms 75B

```
{
  "data": {
    "users": [
      {
        "id": "2",
        "name": "Alice",
        "email": "alice@example.com"
      }
    ]
  }
}
```

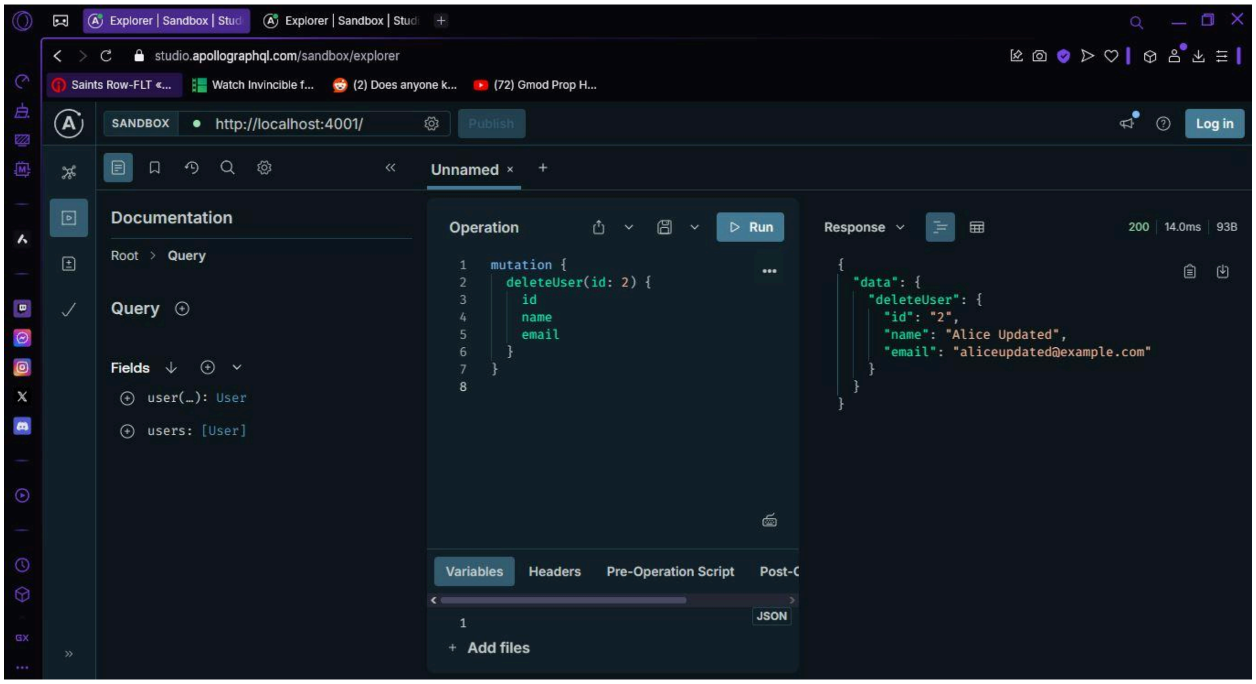
Update Record:



```
mutation {
  updateUser(id: 2, name: "Alice
  Updated", email:
  "aliceupdated@example.com") {
    id
    name
    email
  }
}
```

```
{
  "data": {
    "updateUser": {
      "id": "2",
      "name": "Alice Updated",
      "email": "aliceupdated@example.com"
    }
  }
}
```

Delete Record:

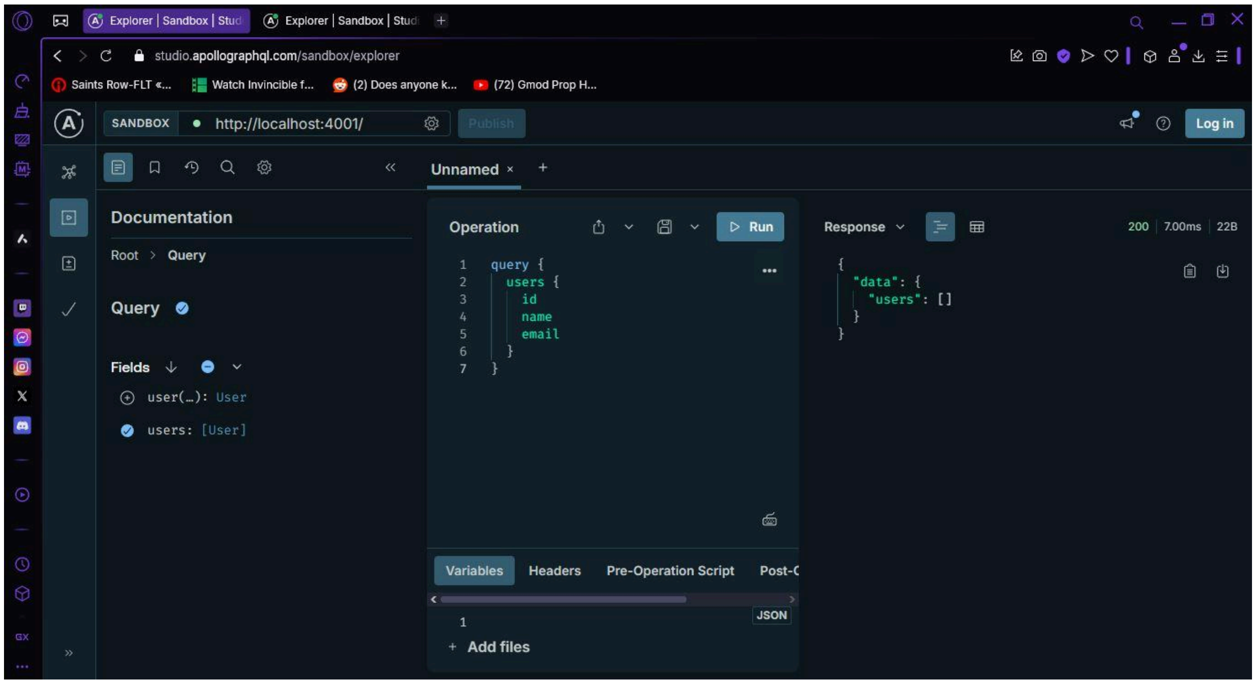


Operation

```
mutation {
  deleteUser(id: 2) {
    id
    name
    email
  }
}
```

Response

```
{
  "data": {
    "deleteUser": {
      "id": "2",
      "name": "Alice Updated",
      "email": "aliceupdated@example.com"
    }
  }
}
```



Operation

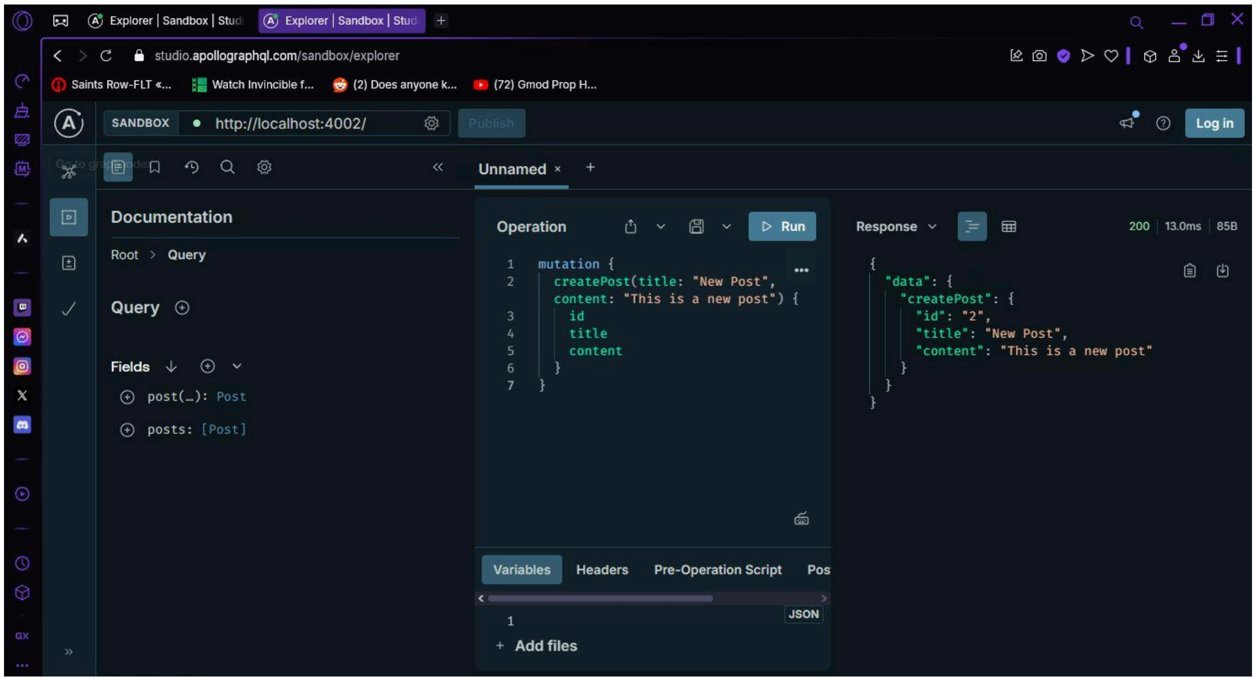
```
query {
  users {
    id
    name
    email
  }
}
```

Response

```
{
  "data": {
    "users": []
  }
}
```

Post Service - Create, Read, Update, and Delete Records Verification:

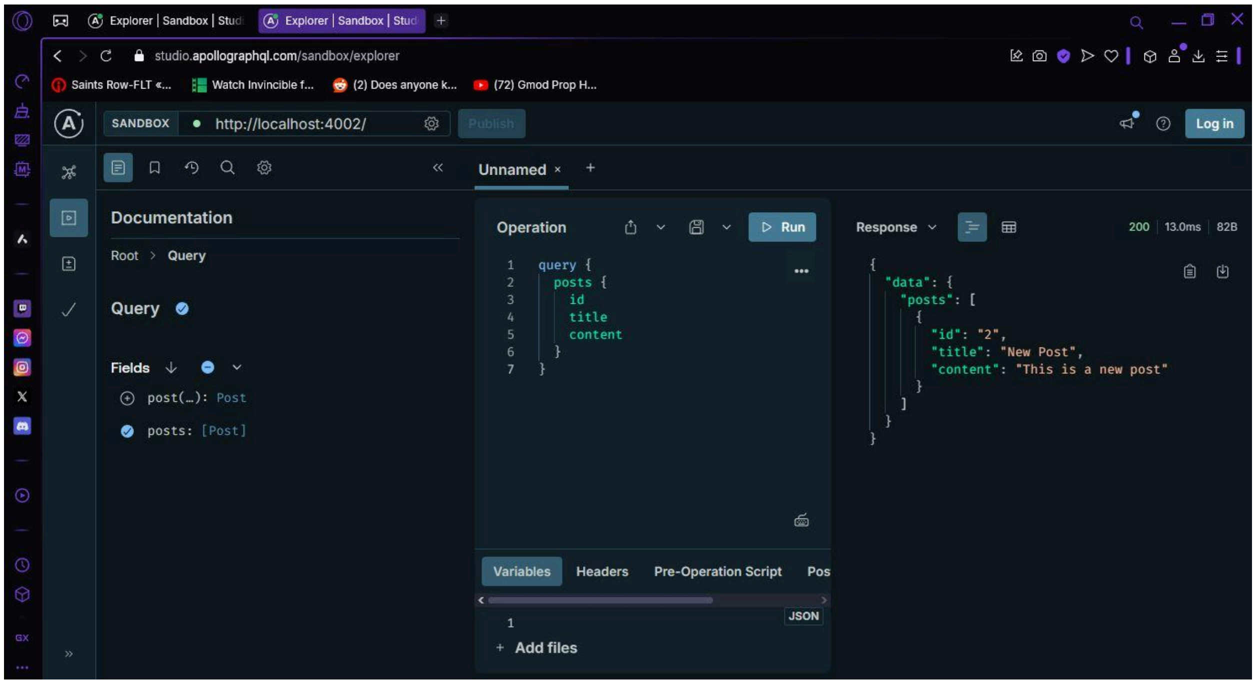
Create Record:



```
mutation {
  createPost(title: "New Post",
  content: "This is a new post") {
    id
    title
    content
  }
}
```

```
{
  "data": {
    "createPost": {
      "id": "2",
      "title": "New Post",
      "content": "This is a new post"
    }
  }
}
```

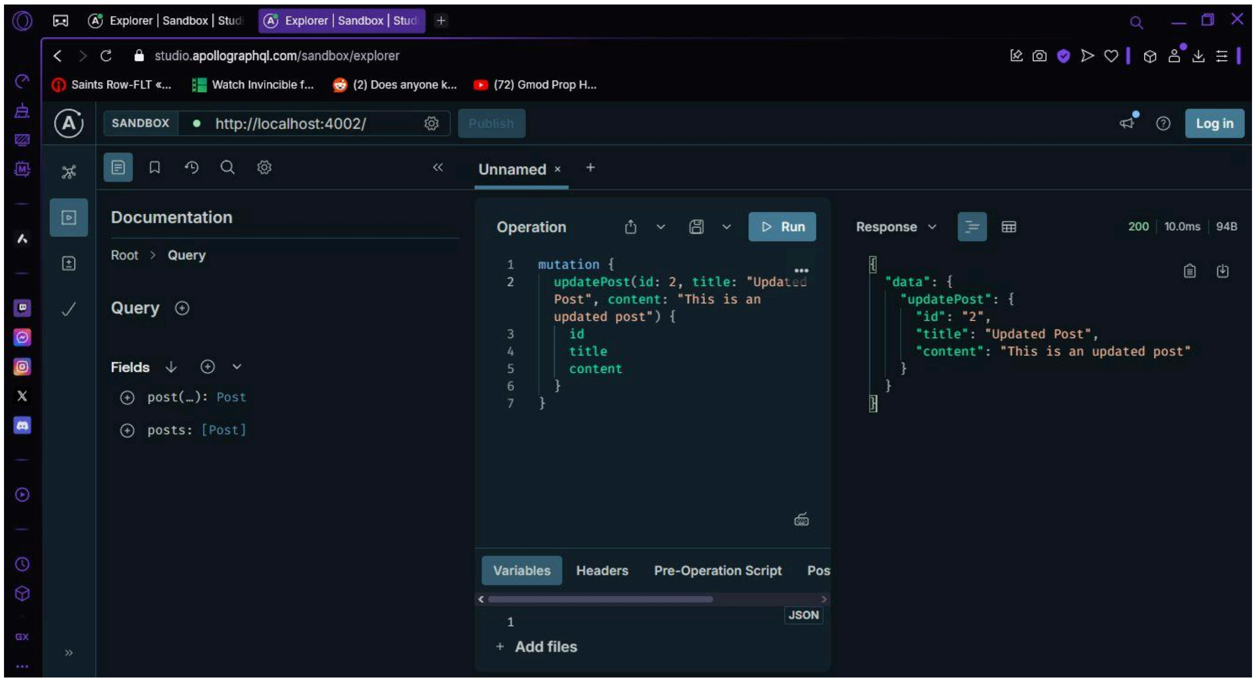
Read Record:



```
query {
  posts {
    id
    title
    content
  }
}
```

```
{
  "data": {
    "posts": [
      {
        "id": "2",
        "title": "New Post",
        "content": "This is a new post"
      }
    ]
  }
}
```

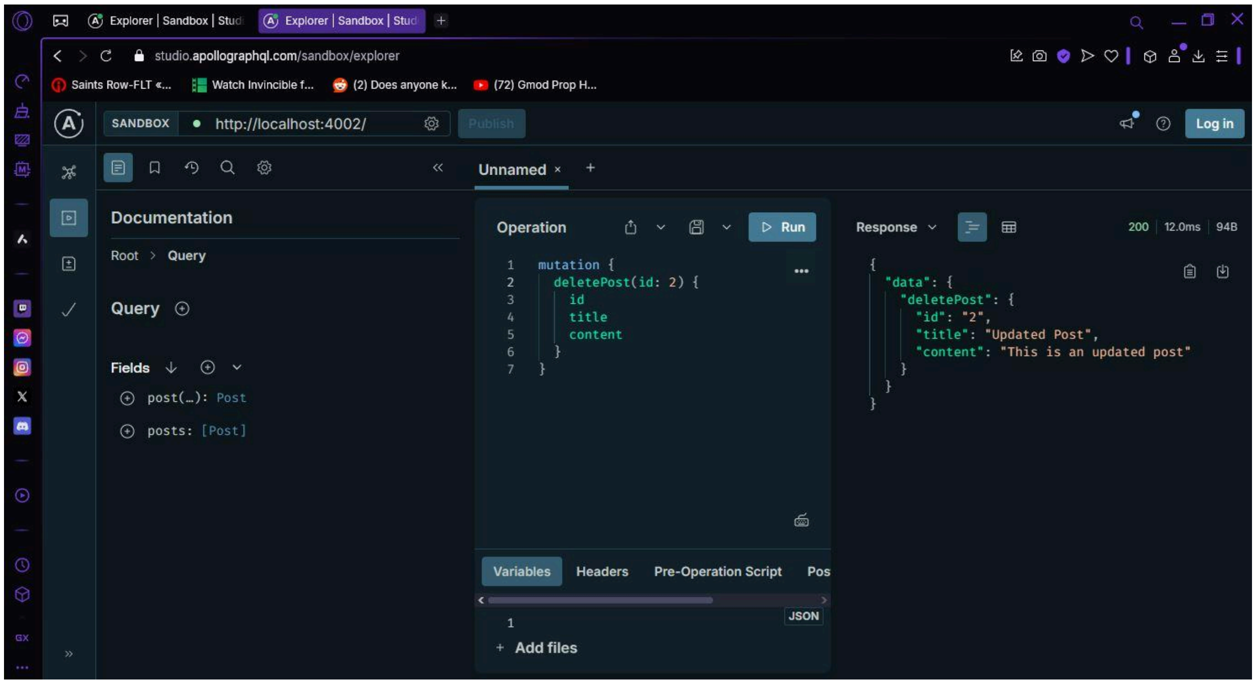
Update Record:



```
mutation {
  updatePost(id: 2, title: "Updated
  Post", content: "This is an
  updated post") {
    id
    title
    content
  }
}
```

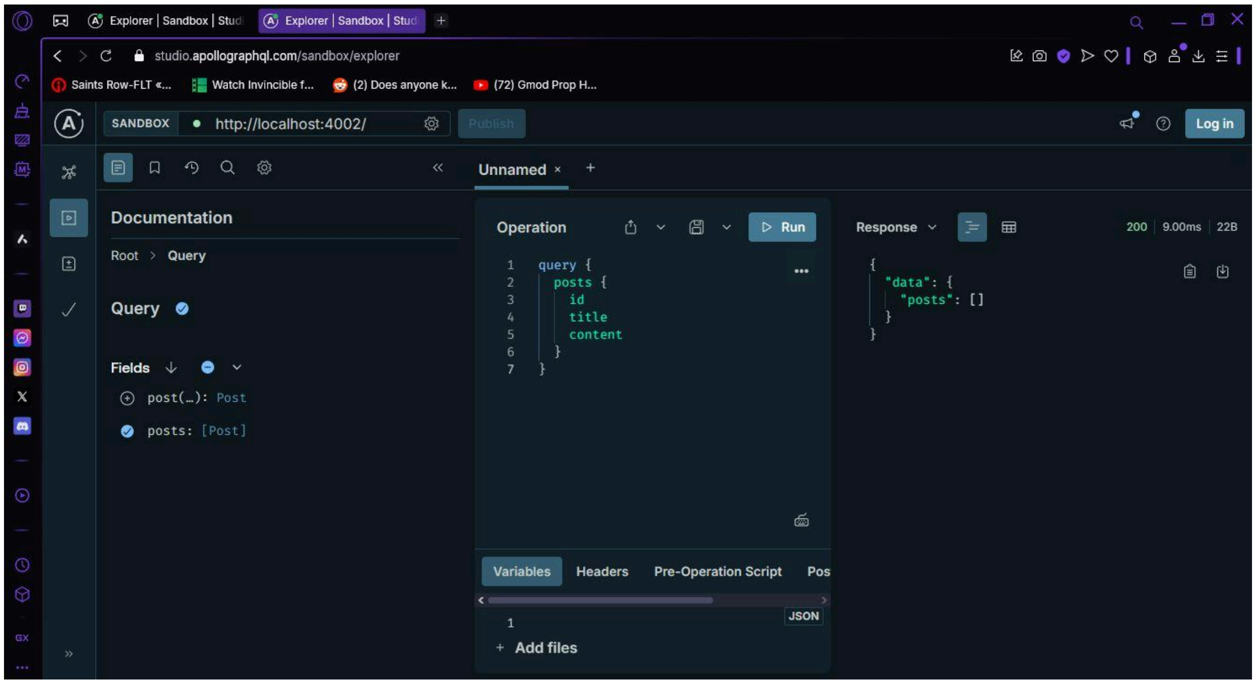
```
{
  "data": {
    "updatePost": {
      "id": "2",
      "title": "Updated Post",
      "content": "This is an updated post"
    }
  }
}
```

Delete Record:



```
mutation {
  deletePost(id: 2) {
    id
    title
    content
  }
}
```

```
{
  "data": {
    "deletePost": {
      "id": "2",
      "title": "Updated Post",
      "content": "This is an updated post"
    }
  }
}
```



```
query {
  posts {
    id
    title
    content
  }
}
```

```
{
  "data": {
    "posts": []
  }
}
```

Short Reflection (3-5 sentences):

- What do database migrations do and why are they useful?
  Database migrations are used to make changes to a database, like adding or changing tables. They help keep the database organized and up to date, and make sure the changes work smoothly across different environments. It's really useful because it helps avoid errors and keeps everything consistent.
- How does GraphQL differ from REST for CRUD operations?
  After doing a bit of research, GraphQL is different from REST because it lets you get exactly the data you need with just one request. With REST, you often need multiple endpoints for different actions, but GraphQL gives you more control and makes things more efficient. It reduces the chances of getting too much or too little data, which makes everything faster and more flexible.