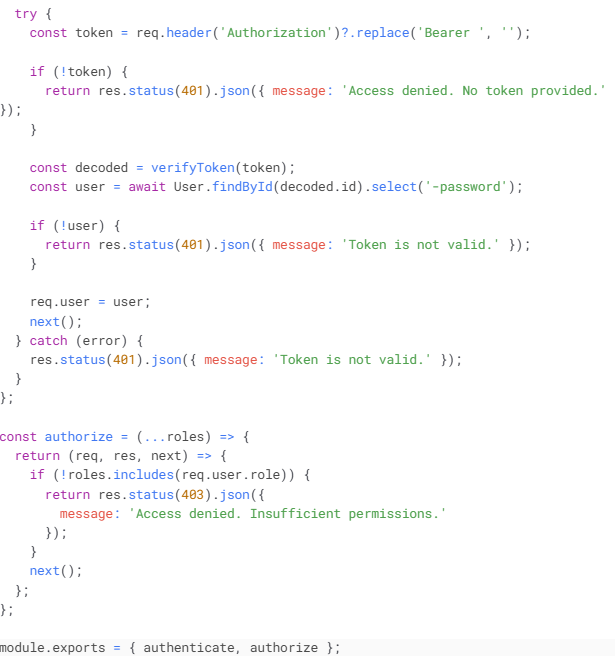
# **Experiment No: 4**

## **Source Code**

### **1. JWT Utility (backend/utils/jwt.js)**

**

### **2. Authentication Middleware (backend/middleware/**[**auth.js**](http://auth.js)**) :**

**

### 

### 

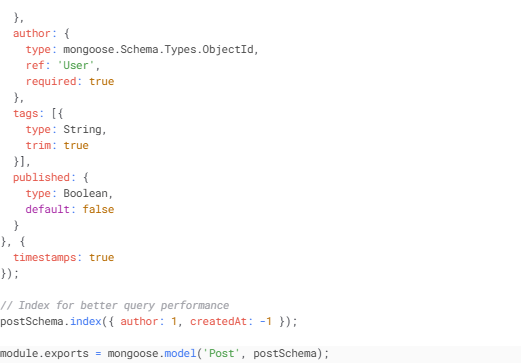
### 

### **3. Authentication Controller (backend/controllers/authController.js)**

**

### **4. Post Model (backend/models/Post.js)**

**

**

### 

### 

### 

### **5. Post Controller (backend/controllers/postController.js)**

**

### 

### 

### 

### 

### 

### 

### 

### **6. Routes**

* Auth Routes (backend/routes/[auth.js](http://auth.js))
* Post Routes (backend/routes/posts.js)  
  

### 

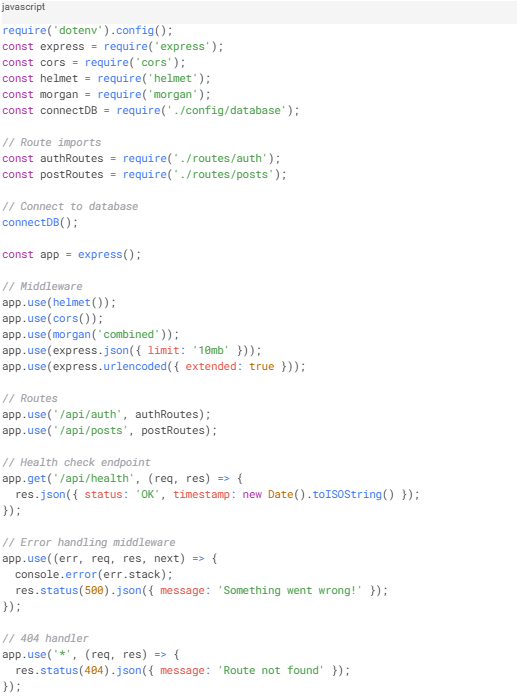
### 

### 

### 

### 

### **7. Server Setup (backend/server.js)**

**

### 

### 

### **8. API Service (frontend/src/services/api.js)**

**

### 

### 

### **9. Auth Service (frontend/src/services/auth.js)**

**

### 

### 

### 

### 

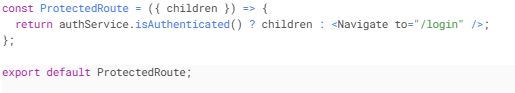
### 

### 

### **10. Post Service (frontend/src/services/posts.js)**

**

### **11. Protected Route Component (frontend/src/components/ProtectedRoute.js)**

**

### 

### 

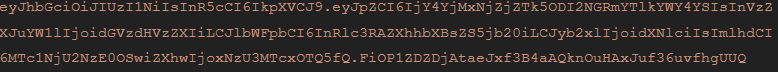
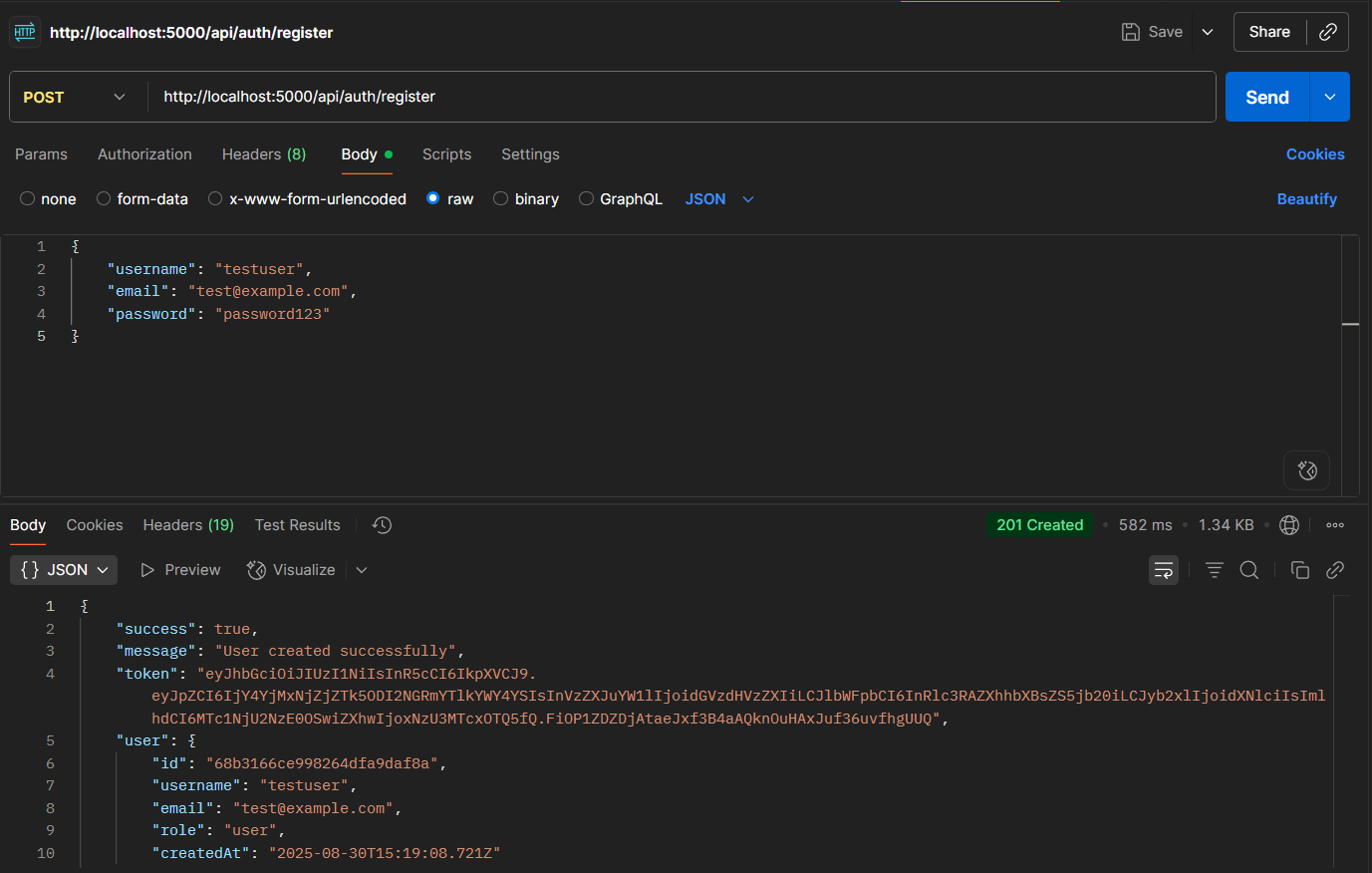
### 

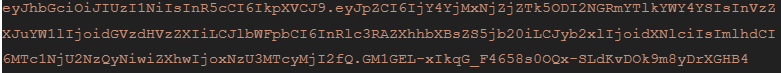
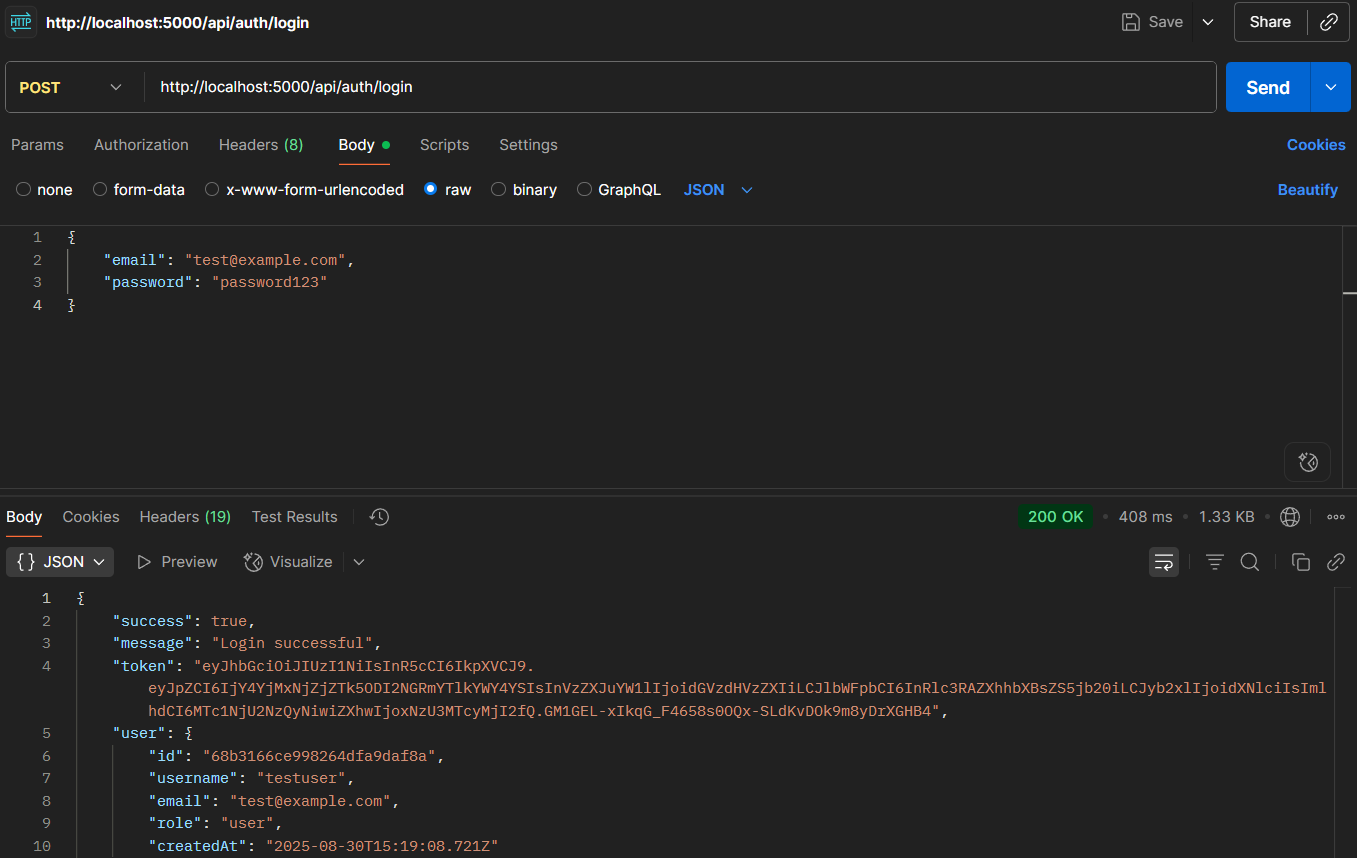
### 

### **12. Login Component (frontend/src/components/Login.js)**

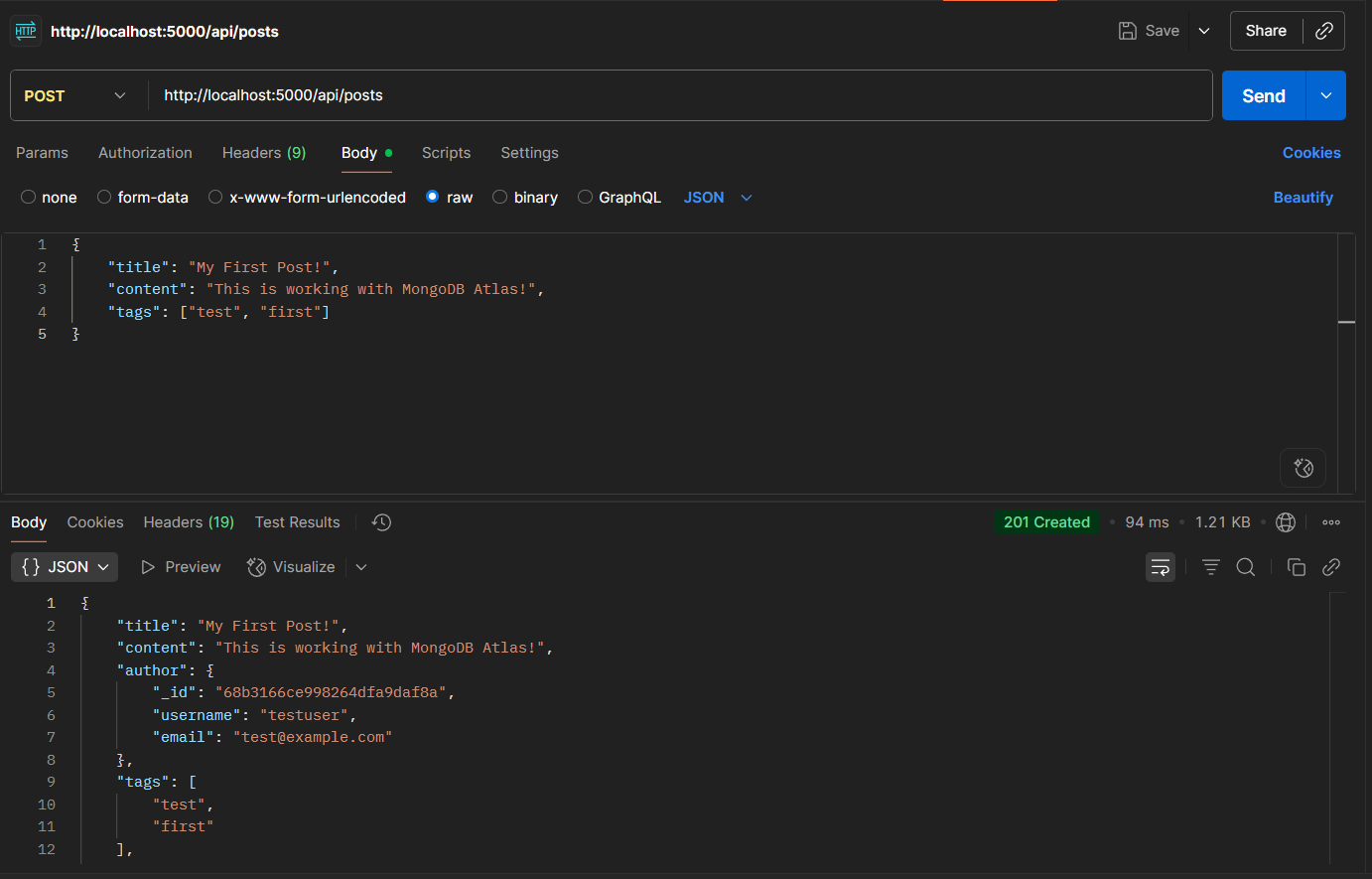
**

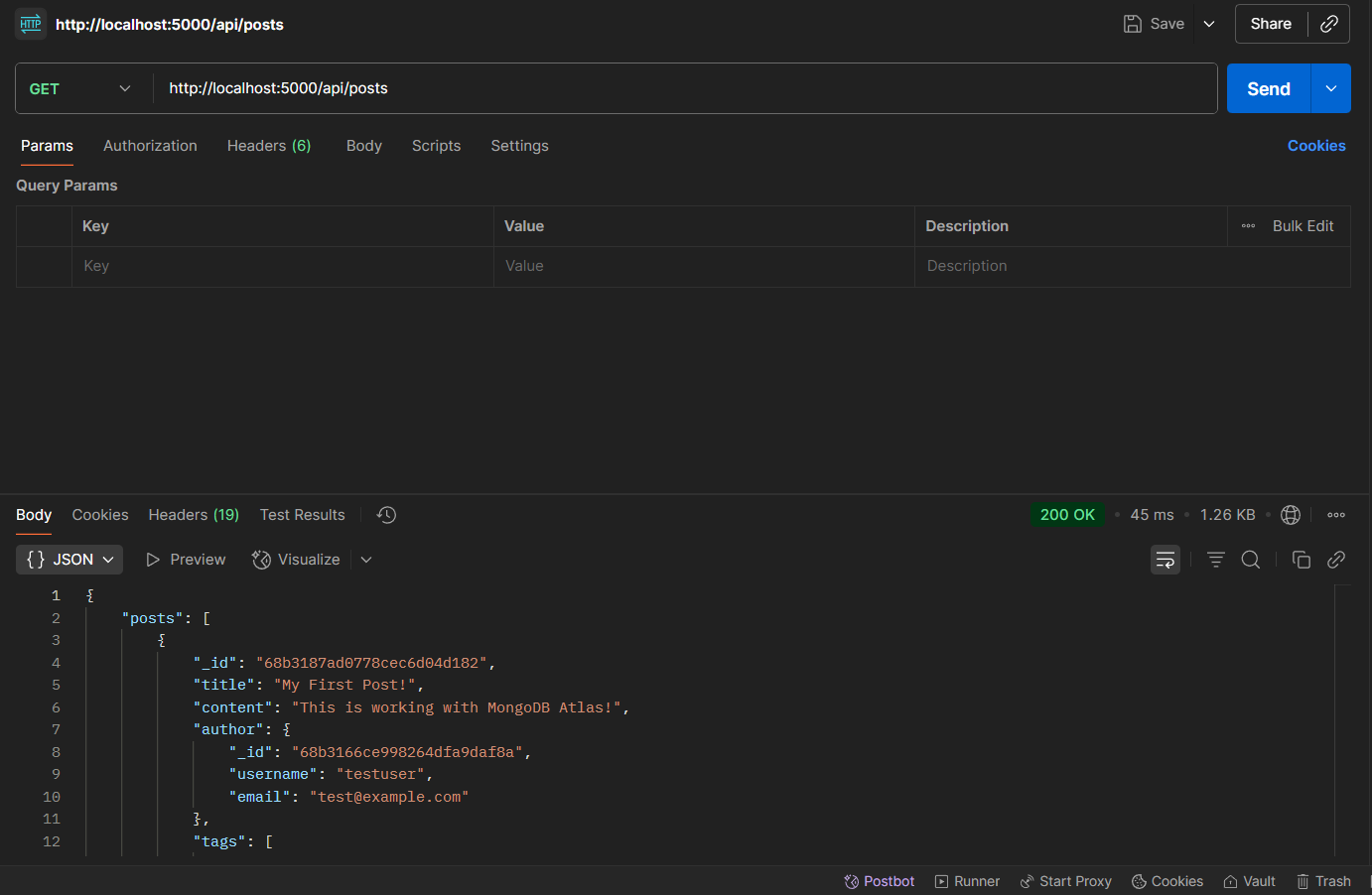
## **Output**

 *Fig 4.1* ***User Registration*** *– New users can register with username, email, and password.*

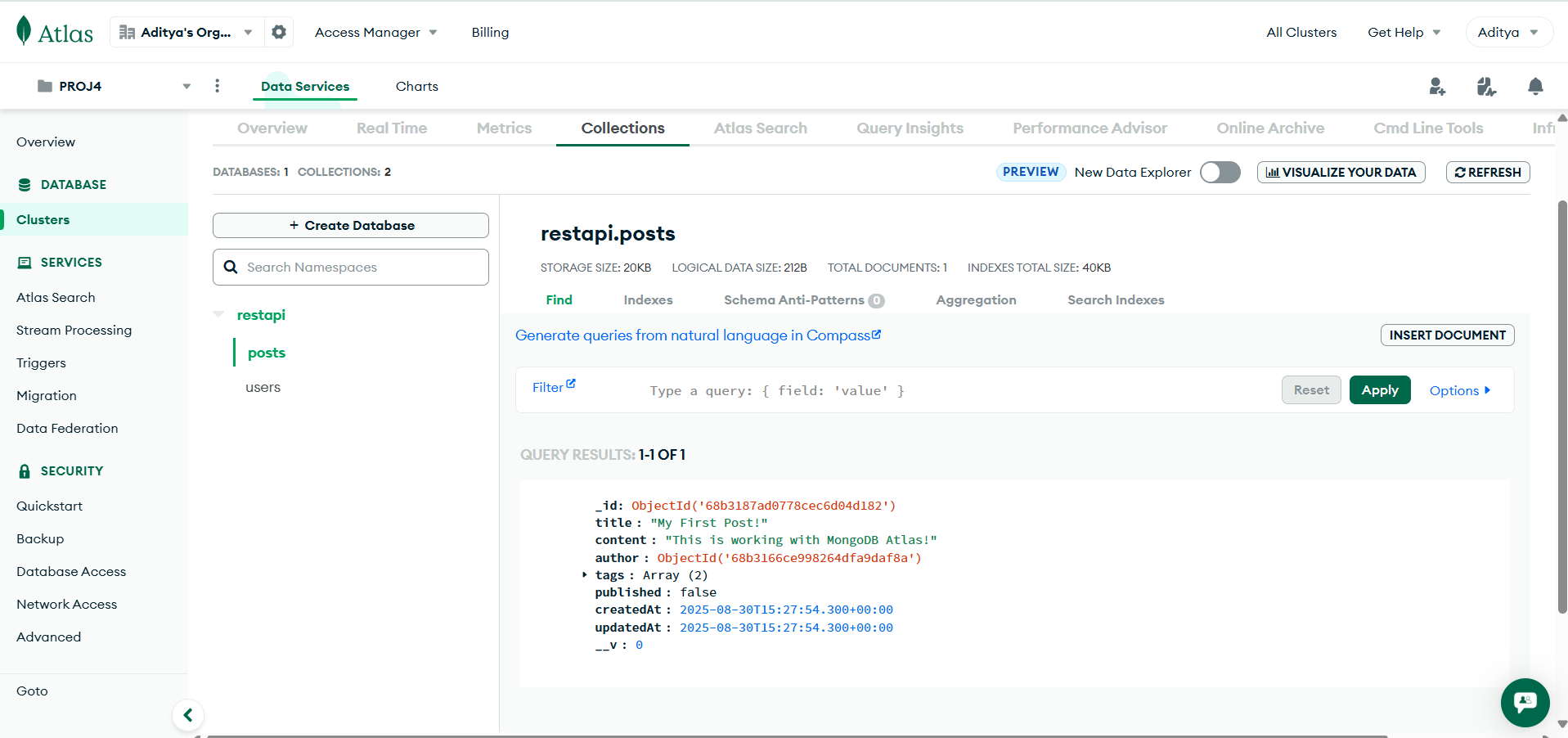


*Fig4.2* ***Login & Token Generation*** *– Valid login returns a JWT token.*

 *Fig4.3* ***Create Post*** *– Authenticated user creates a post.*



*Fig 4.4* ***Fetch Posts with Pagination*** *– Posts retrieved with pagination and author details.*

 *Fig 4.5* ***Frontend Integration*** *– Login and protected route handling on frontend.*