**Full Stack Development with MERN**

**API Development and Integration Report**

| Date | 08/07/2024 |
| --- | --- |
| Team ID | SWTID1720278137 |
| Project Name | Online Learning Platform |
| Maximum Marks |  |

**Project Title:** **Online Learning Platform (EduVerse)**  
**Date: 08/07/2024**  
**Prepared by:** SWTID1720278137

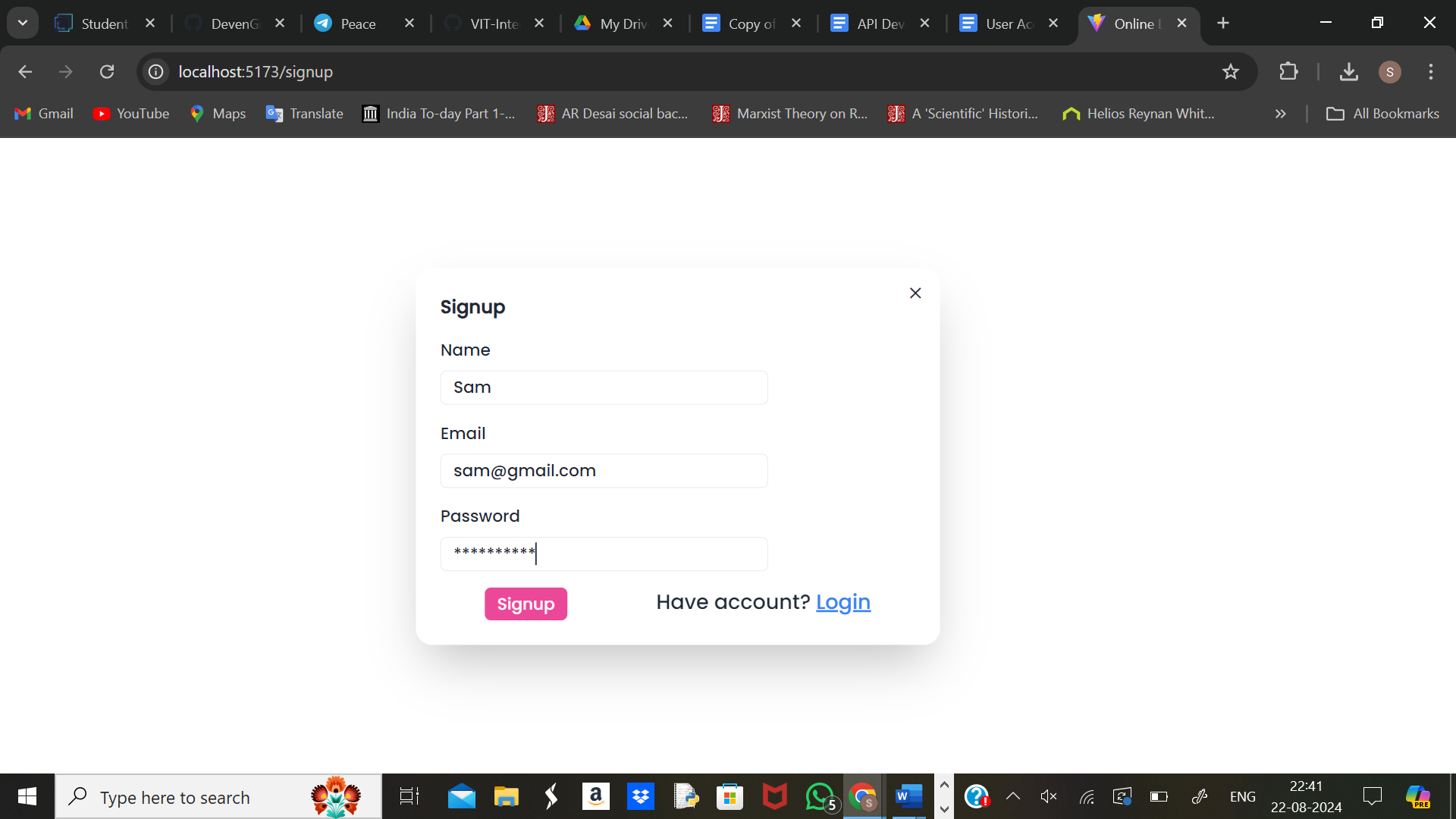
**Objective**  
The objective of this report is to document the API development progress and key aspects of the backend services implementation for the Online Learning Platform (EduVerse) project.

**Technologies Used**

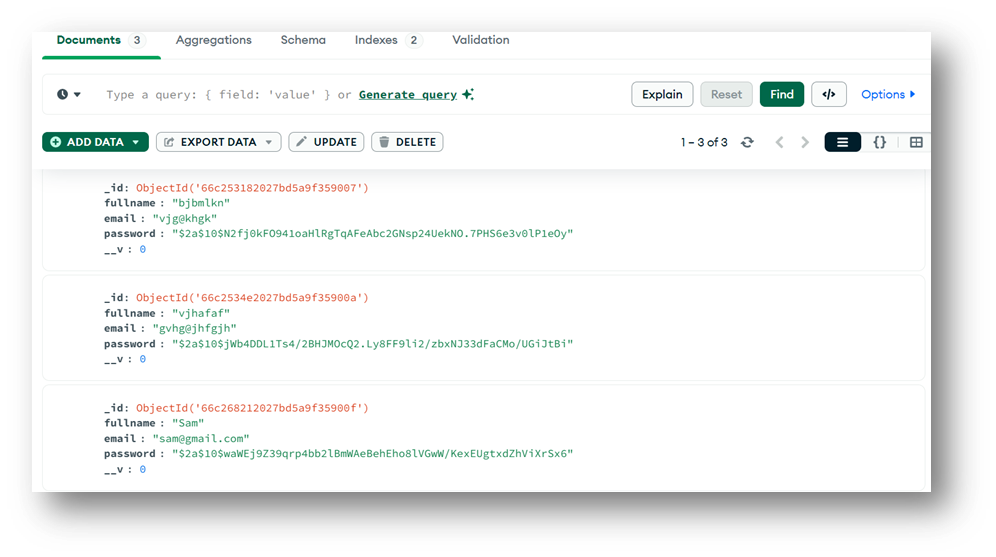
* **Backend Framework:** Node.js with Express.js
* **Database:** MongoDB
* **Authentication:** None

**Project Structure:**

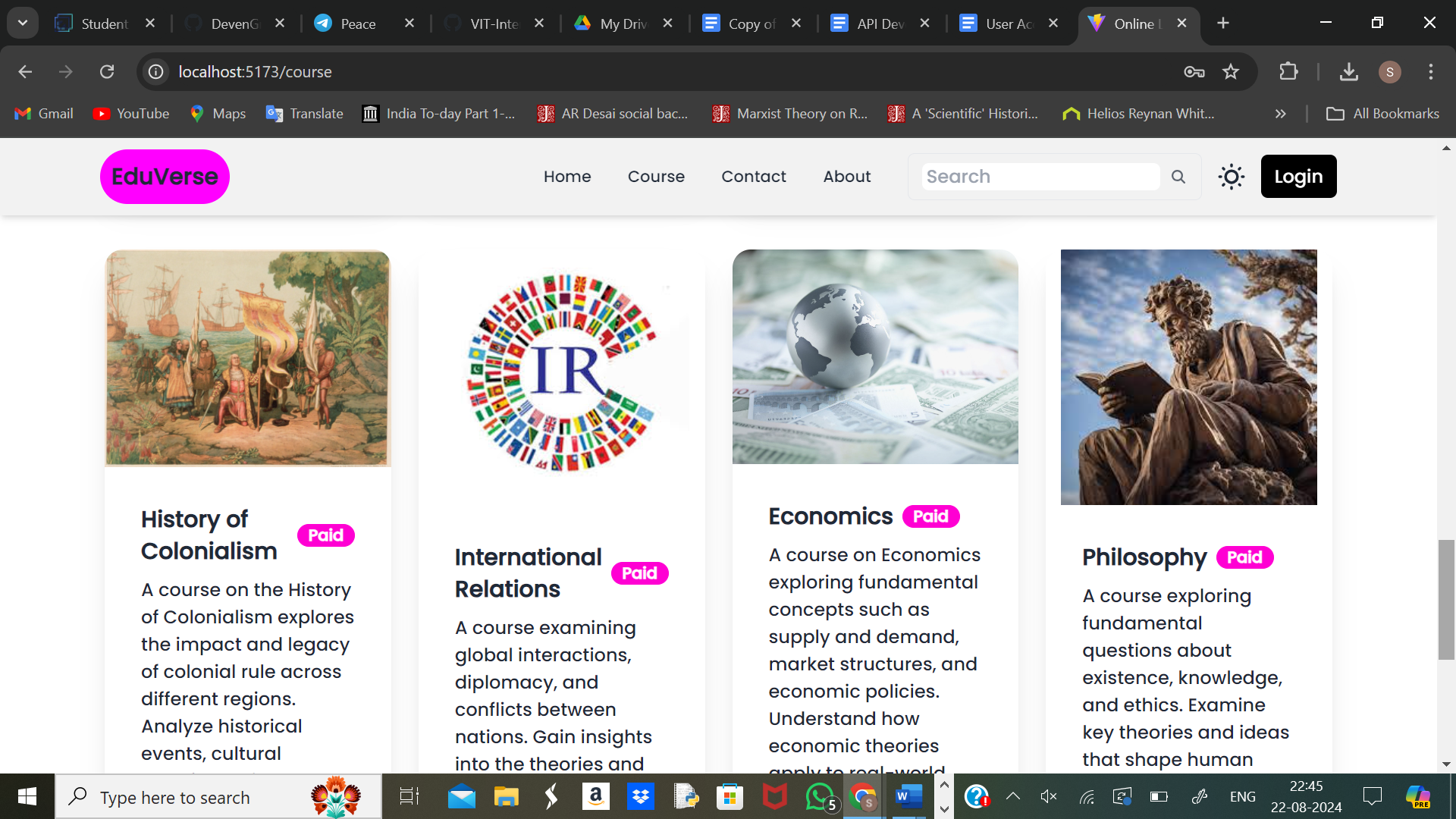
Registration by user:



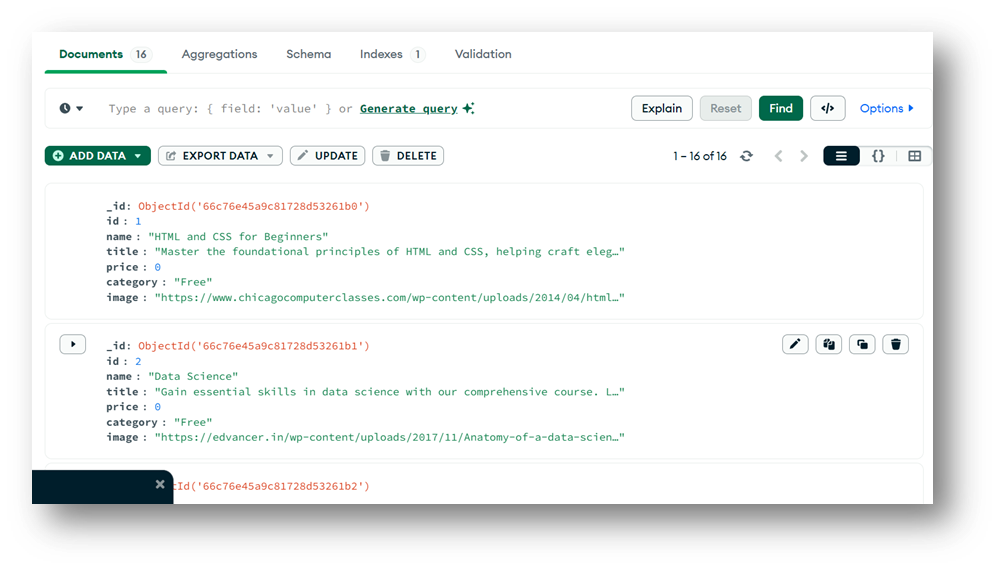
Updated MongoDB user collection upon Signup:

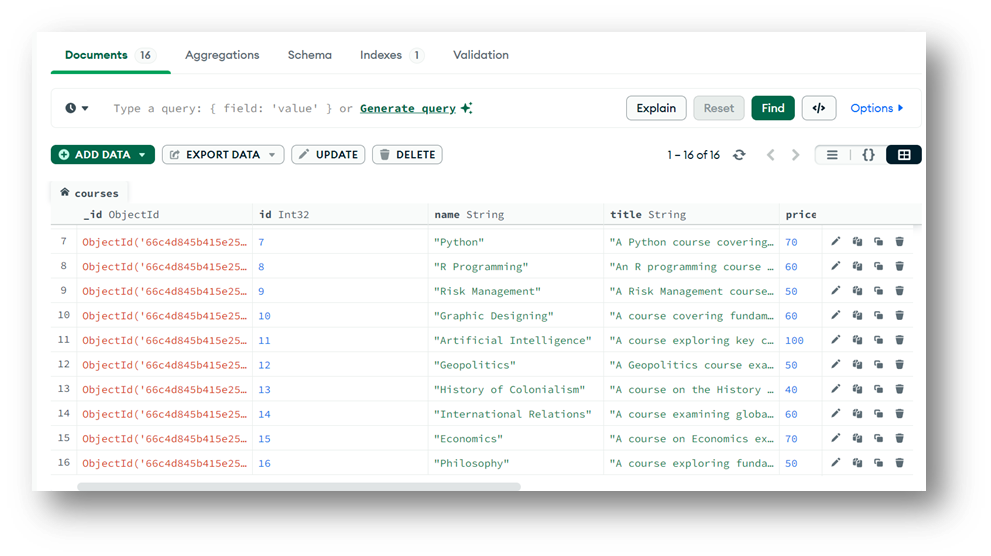


Course page visible upon successful user login:



Information stored in course collection:





**Key Directories and Files**

1. **/controllers**

* **course.controller.js**: Manages all operations related to courses, including creation of new course, retrieving and fetching list of courses available based on ID, and updating or deleting course details.
* **user.controller.js**: Manages all operations related to users, including user login, registration, updating user profile, and deleting user account.

1. **/models**

* **course.model.js**: Defines the schema and structure for the course data in the database.
* **user.model.js**: Defines the schema and structure for the user data in the database.

1. **/routes**

* **course.route.js**: Defines the API endpoints related to course management on the platform. Maps HTTP requests (GET, POST, PUT, DELETE) to the corresponding functions in course.controller.js, facilitating operations like creating, updating, retrieving, and deleting courses.
* **user.route.js**: Defines the API endpoints related to user management on the platform. Maps HTTP requests to the appropriate functions in user.controller.js, enabling operations like user registration, login, profile management, and account deletion.

1. **/config**

* **.env file:** Stores sensitive information and environment-specific settings like database credentials, API keys, and port numbers.

**API Endpoints**  
A summary of the main API endpoints and their purposes:

**User Authentication**

* **POST /api/user/register** - Registers a new user.
* **POST /api/user/login** - Authenticates a user and proceeds to the course page upon successful login.

**User Management**

* **GET /api/user/-** Retrieves user information based on their ID.
* **PUT /api/user/**- Updates user information such as password, username etc. based on ID.

**Course Management**

* **POST /api/add\_courses** - Adds new courses with relevant details.
* **POST /api/modify\_courses** - modifies course details such as title, pricing, description etc.
* **POST /api/delete\_courses** - Deletes specified course and all its attributes.
* **POST /api/get\_courses-** Retrieves course details based on search input given by the user.

**Integration with Frontend**  
The backend communicates with the frontend via RESTful APIs. Key points of integration include:

* **User Authentication:** Tokens are passed between frontend and backend to handle authentication.
* **Data Fetching:** Frontend components make API calls to fetch necessary course data for display and interaction.

**Error Handling and Validation**  
Describe the error handling strategy and validation mechanisms:

* **Error Handling:**
* Error responses are sent with appropriate status codes (e.g., 100 for unsuccessful operations, 200 for success).
* Try-catch blocks are used to handle errors in async functions.
* **Validation:**
* Basic validation is performed using conditional checks.
* For example, checking if a user already exists before saving a new user.

**Security Considerations**  
Outline the security measures implemented:

* **Authentication:** It is handled through email and password verification for logging into account and accessing the course section.
* **Database Connection**: Secure connection string is used to connect to MongoDB.