**Project**

**JSPR-Techies**

**Checkpoint 1  
  
11/7/2023**

**Members**

Jayani Sumanka Gerine

Pallavi Dabade

Raksha Varahamurthy

### Surya Subramani

**Table of Contents**

Project Title & Members……………………………….…………….…….1

List of User Stories Completed……………………….…………….…….3

Completed Functionality………………………………….………….…….3

Changes to the User Stories…………………………..….….…………………..6

Breakdown of the existing User Stories…………….…….….…………………7

New User Stories………………………………………….……..……………….9

List of the User Stories Yet to Implement……………..……………………….10

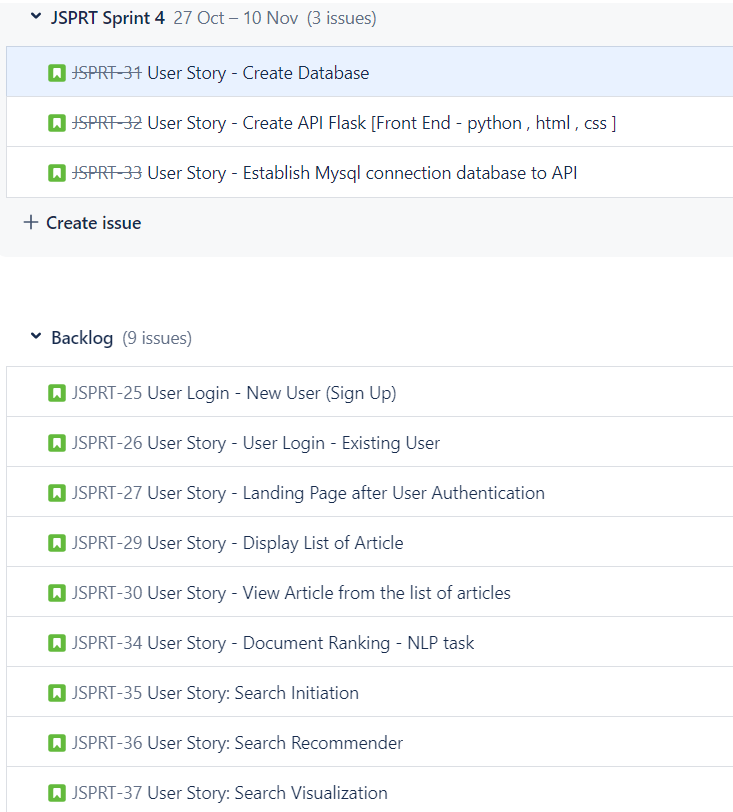
New or Updated User Stories………………………………..…………………11

**Iteration Updates - 10/27/23 to 11/10/23**

**List of User Story Completed**

(List the user stories that you have implemented at this iteration. (1 pt))

1. User Story - Create Database
2. User Story - Create API Flask [Front End - python , html , css ]
3. User Story - Establish Mysql connection database to API

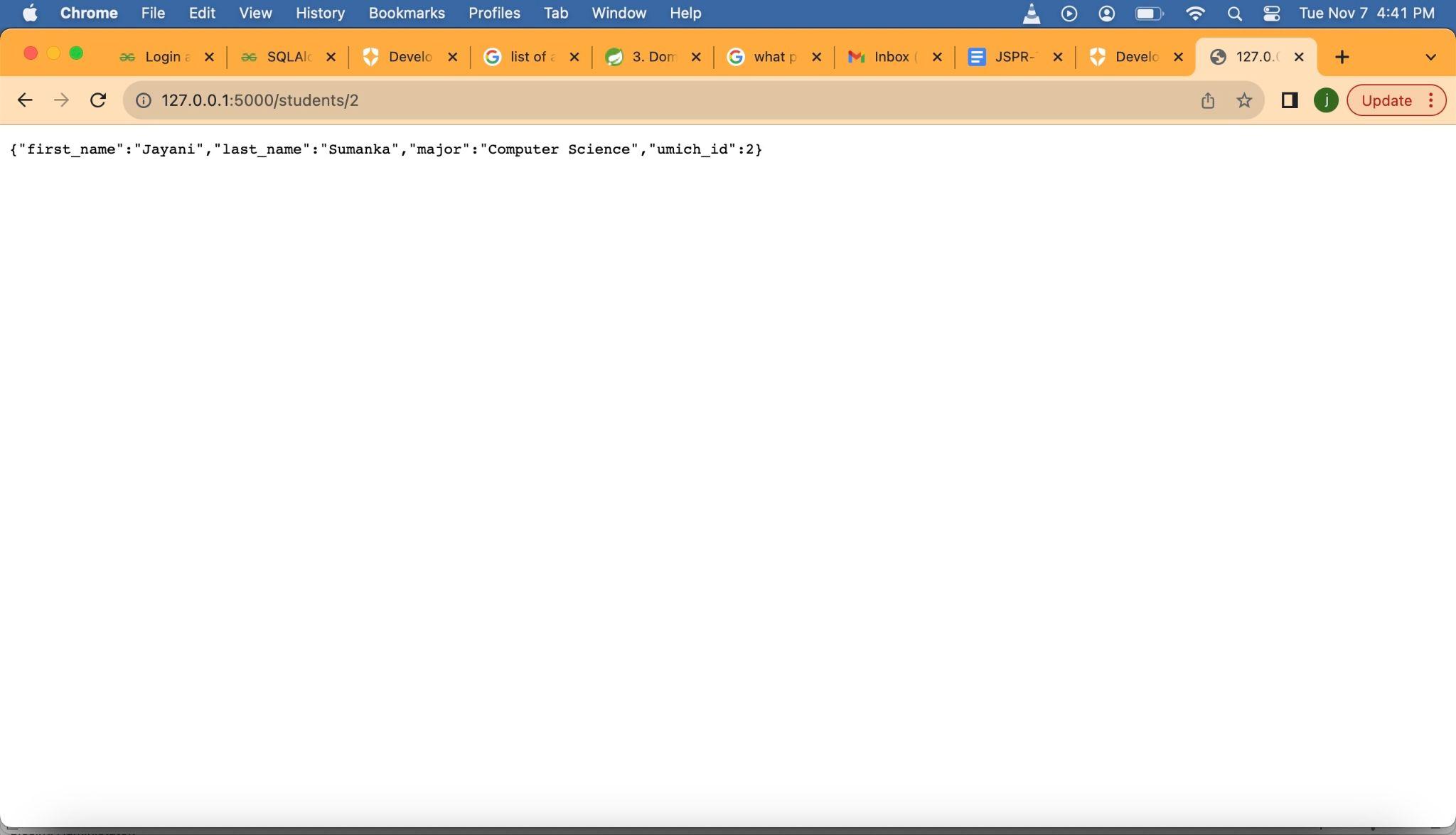


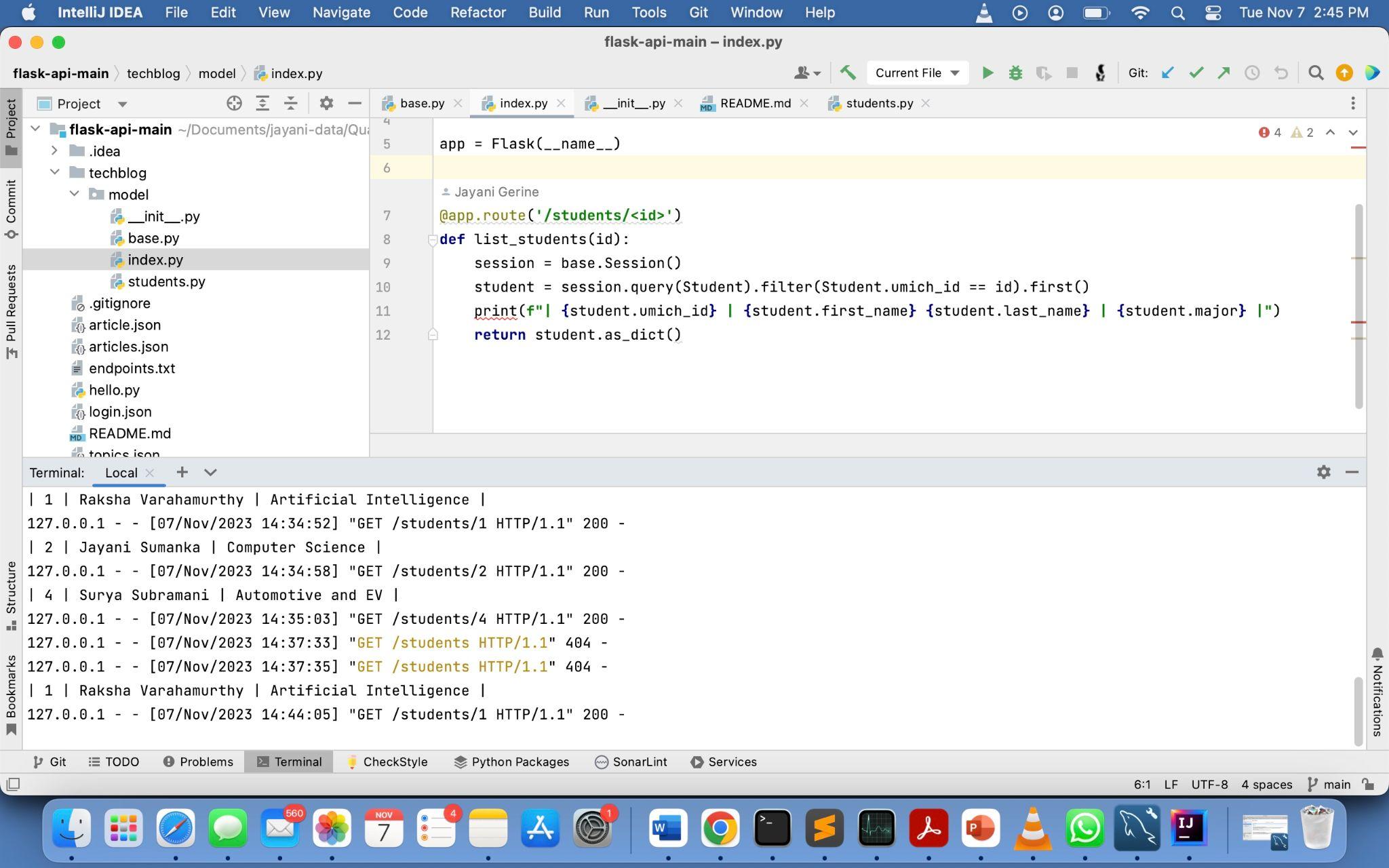
**Completed functionality of the system at the end of iteration**

(What functionality does the system have at the end of this iteration? (2 pts))

* The project framework and database foundation will be ready.
* Database connection to API will be established.

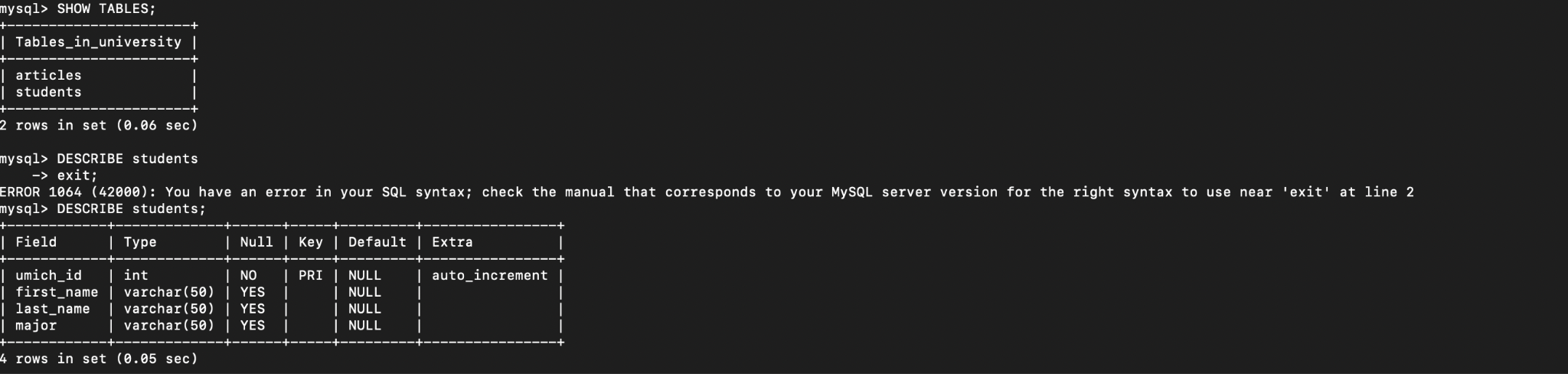
**Screenshots of the functionality achieved at the end of this iteration**

The Flask API retrieves the details of a student saved in the database after establishing the Database connection from the API.

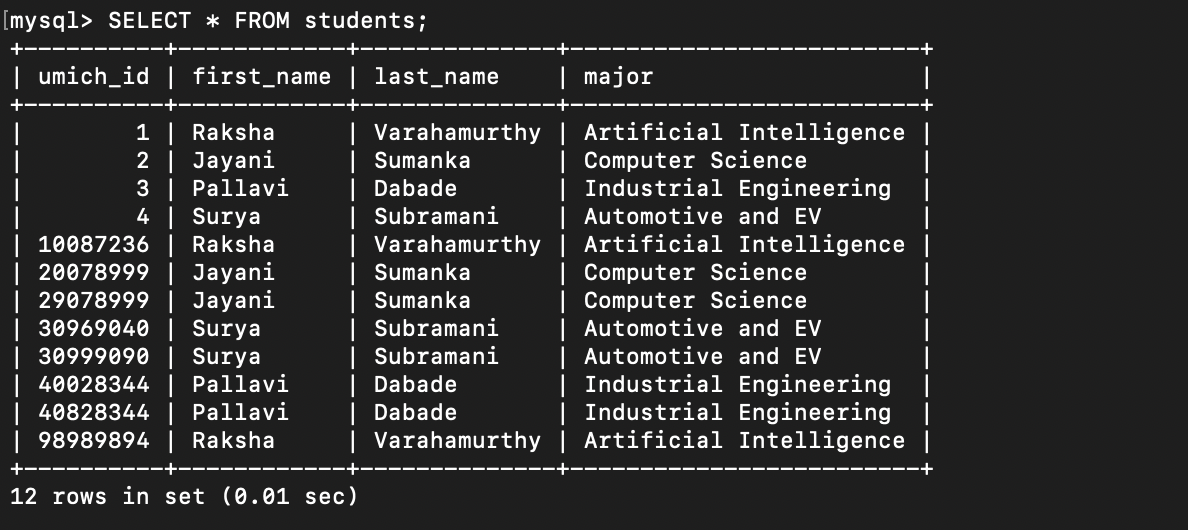


Database Table Screenshots

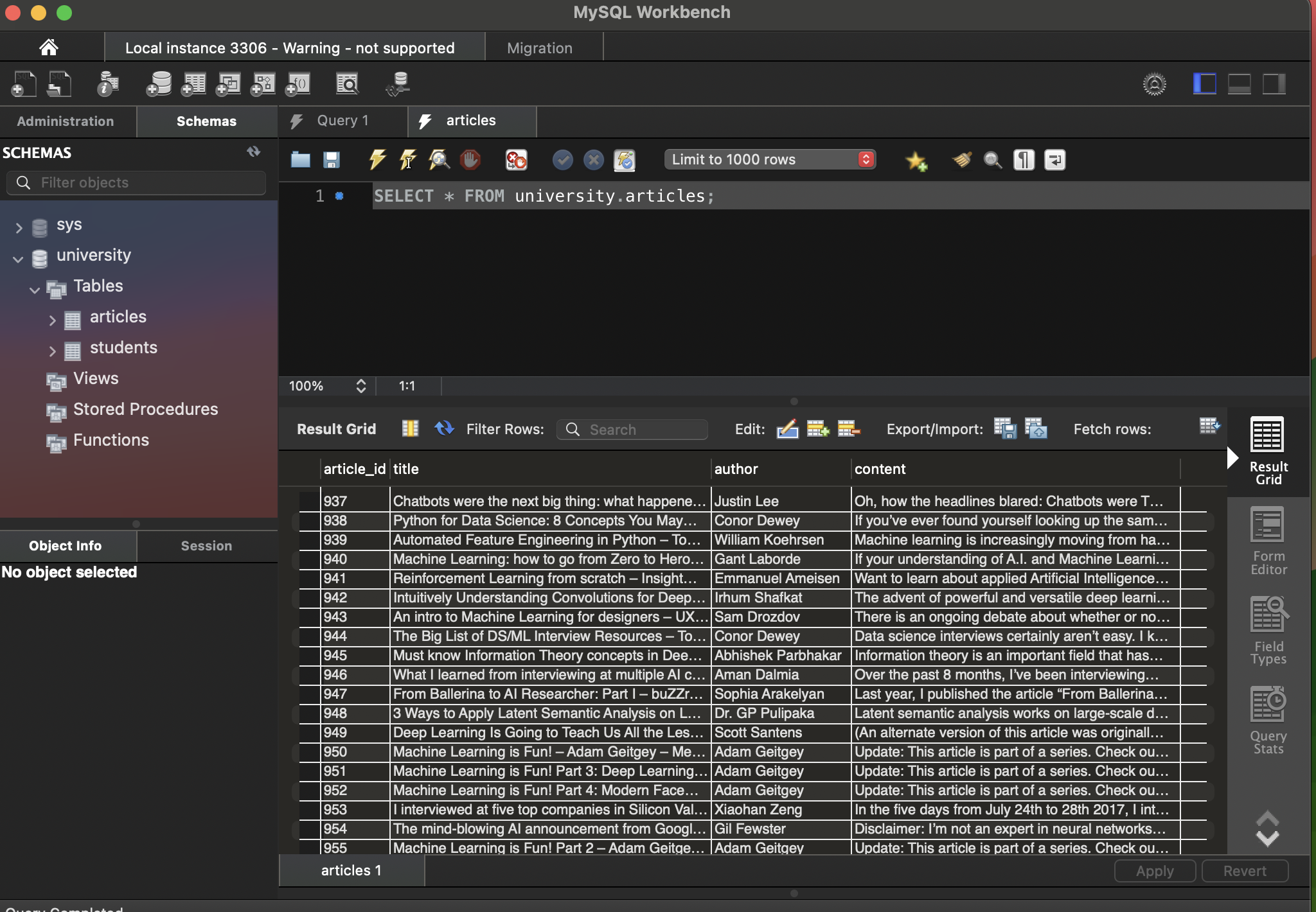
Created “University” Database with “student” and “article” table.



Student Table



Article Table



**Changes to the User Stories**

We have made some changes to your list of user stories to provide more clarity and structure. Here are the modified user stories

We have made some changes to our list of completed user stories to provide more clarity and structure. Here are the modified user stories

| Use Case 1 | Create Database |
| --- | --- |
|  | Description |
| Actors | Developer |
| Triggers | Technical requirement |
| Precondition | • Database model define |
| Post Conditions | • Retrieve the article data from database. |
| Normal Operations | • The database and tables are successfully created. |
| Exception | • If there are issues with the database creation process, such as connectivity or permissions, the database may not be successfully created.  • Technical issues during database setup could lead to failure. |

6

| Use Case 2 | Create API Flask |
| --- | --- |
|  | Description |
| Actors | Developer |
| Triggers | • Application Programming Interface where users can log in and access the articles. |
| Precondition | • Use Flask to build an API |
| Post Conditions | • Users can access on frontend the articles. |
| Normal  Operations | • Users can access articles on the frontend using the API |
| Exception | • If there are errors in the API development or configuration, users may not be able to access articles.  • Technical issues with the API could lead to its unavailability. |

| Use Case 3 | Establish MySql Connection database to API |
| --- | --- |
|  | Description |
| Actors | Developer |
| Triggers | • To search the user defined keyword in database |
| Precondition | • Article table defined |
| Post Conditions | • Established connection will allow users to interact with application and access the articles |
| Normal  Operations | • Users can search for user-defined keywords in the database via the API. |
| Exception | • If there are database connection issues, such as incorrect credentials or server downtime, users won't be able to access articles.  • Technical issues with the MySQL connection can disrupt this functionality |

**Breakdown of the existing User Stories**

Did you break down any further user stories?

We have created a generalized user story encapsulates the front-end functionality for the JSPR-Techies blog, including user registration, login, landing page display, article search, display, and reading, with triggers, normal operations, and exceptions for each sub-story.

**User Story - Front-End Functionality for JSPR-Techies Blog**

• **Size**: Large (Composed of sub-stories)

• **User Role**: Student, Faculty, and Alumni of University of Michigan

• **Goal**: Provide a user-friendly and seamless front-end experience for users of the JSPR Techies blog, facilitating user registration, login, article search, display, and reading. • **Reason**: To enable users associated with the University of Michigan to access and interact with the JSPR-Techies blog with ease.

• **Preconditions**: Users have internet access, a web browser, and are connected to the University of Michigan domain.

• **Post conditions**: Users can effectively navigate the JSPR-Techies blog, creating accounts, logging in, searching for articles, viewing a list of articles, and reading articles.

**Sub-Stories**:

1. **User Registration and Account Creation**

o **Trigger**: New users accessing the JSPR-Techies blog.

o **Normal Operations**:

▪ New users land on the login page.

▪ Users create accounts using unique usernames and their email

addresses as passwords.

o **Exception**:

▪ If the domain name is not recognized or the registration process

encounters errors, users may not be able to create accounts.

2. **User Login**

o **Trigger**: Existing users accessing the JSPR-Techies blog.

o **Normal Operations**:

▪ Existing users land on the login page.

▪ Users enter their unique usernames and email addresses to log in.

o **Exception**:

▪ If the entered credentials do not match, users won't be able to log in.

3. **Landing Page After User Authentication**

o **Trigger**: Users successfully logging in.

o **Normal Operations**:

▪ Authenticated users land on the JSPR-Techies blog's home page and see a welcome message.

o **Exception**:

▪ If users enter incorrect credentials or fail to authenticate, they won't reach the landing page.

4. **Article Search and Retrieval**

o **Trigger**: Users initiating article search by entering keywords and clicking the submit button.

o **Normal Operations**:

▪ Users search for relevant articles by entering keywords.

▪ Users receive search results, with the ability to view the top 5 articles based on their preferences.

o **Exception**:

▪ If the search query returns no results, users will receive a "no results found" message.

▪ Technical errors or server issues may disrupt the search functionality.

8

5. **Display List of Articles**

o **Trigger**: Users entering keywords and clicking enter.

o **Normal Operations**:

▪ Users enter keywords, initiate a search, and receive a list of articles

based on the search criteria.

o **Exception**:

▪ Technical issues or database errors may prevent the list of articles from loading.

6. **View Article from the List of Articles**

o **Trigger**: Users clicking on a specific article in the list.

o **Normal Operations**:

▪ Users click on a specific article to read it.

o **Exception**:

▪ If the selected article is unavailable or removed, users won't be able to read it.

▪ Technical issues or server errors may disrupt the article reading

functionality.

This generalized user story encapsulates the front-end functionality for the JSPR-Techies blog, including user registration, login, landing page display, article search, display, and reading, with triggers, normal operations, and exceptions for each sub-story.

**New User Stories**

Did you identify any new user stories during this iteration and, if so, did you add them to the product backlog or decide to implement them right away? Explain. (2 pts)

**User Story: Redirect to Login Page or Article Page**

• **User Role**: General User

• **Goal**: To provide users with the option to navigate to either the login page or the article page based on their preference or intent.

• **Reason**: To ensure that users have the either flexibility to log in or directly access articles depending on their needs and interests.

**Scenario**:

1. **Trigger**: The user accesses the JSPR-Techies blog.

2. **Normal Operations**:

o The user lands on the JSPR-Techies blog homepage.

o The homepage displays two prominent options:

▪ "Log In" - For users who want to access their accounts or create new ones?

▪ "Explore Articles" - For users who wish to directly browse and read

articles without logging in.

o The user selects one of the two options:

▪ If the user selects "Log In," they are redirected to the login page.

▪ If the user selects "Explore Articles," they are redirected to the article page.

o The user can then proceed with their chosen action, either logging in or reading articles.

9

3. **Exception**:

o If there are technical issues or server errors preventing the display of the homepage or the two options, the user may encounter difficulties in selecting their intended action.

o In case the user attempts to access restricted content without logging in, they will be prompted to log in before proceeding.

**Lesson Learned**

What are the "lessons learned" at the end of this iteration? What would you do differently next time? Explain. (2 pts)

**Clearer API Specification and Documentation**:

Lesson Learned: It's crucial to have a well-documented API specification that defines the endpoints, request/response formats, and authentication methods.

What We'd Do Differently: In the next iteration, we would invest more time in creating clear and comprehensive API documentation. This would include using tools like Swagger or OpenAPI to document the API endpoints and their functionality.

**Testing and Validation**:

Lesson Learned: Rigorous testing is essential to catch and address potential issues early in the development process. This includes unit testing, integration testing, and validating input data.

What We'd Do Differently: In future iterations, we would allocate more resources and time to testing. Automated testing scripts and comprehensive test cases would be implemented to ensure the API's reliability and robustness.

**Clear and Comprehensive Schema Definition**:

Lesson Learned: It is crucial to have a clear and comprehensive schema definition before creating tables. The schema should outline the structure, relationships, and constraints of the database, helping to prevent issues during table design.

What We'd Do Differently: In the next iteration, we would invest more time in thoroughly defining the schema, including entity-relationship diagrams and data dictionaries, to ensure a well structured and organized database.

**Normalization and Data Integrity**:

Lesson Learned: Ensuring data integrity through proper normalization is essential. Normalization reduces data redundancy and minimizes the risk of inconsistencies and anomalies in the database.

What We'd Do Differently: In future iterations, we would apply normalization principles rigorously during the design phase. This would include identifying functional dependencies and adhering to the rules of the normal forms, such as 1NF, 2NF, and 3NF.

10

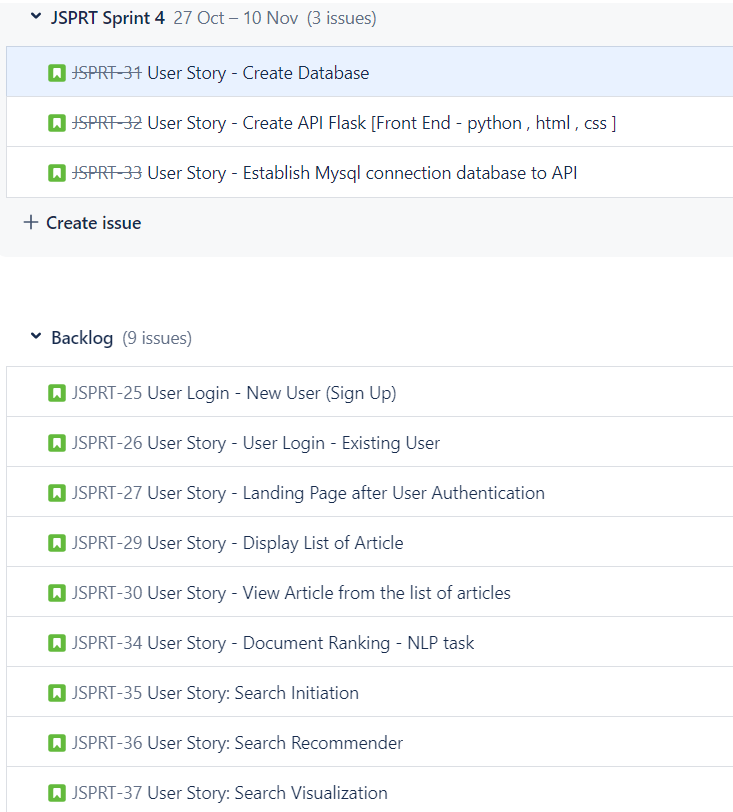
**Constraints and Validation Rules**:

Lesson Learned: Constraints and validation rules are essential for enforcing data integrity. They help prevent invalid or inconsistent data from entering the database.

What We'd Do Differently: In the next iteration, we would define and implement a comprehensive set of constraints, including primary keys, foreign keys, unique constraints, and check constraints. We would also invest in validating data at the application level to ensure it adheres to the defined rules.

**List of the User Stories Yet to Implement**

* Provide an updated numbered list of all user stories yet to be implemented; indicate pre- and post-conditions. (1 pt)



**New or Updated User Stories**

We'll also describe the functionality that the partially implemented system will have at the end of this iteration. Since this isn't the last iteration, we'll focus on the next steps for the project.

11

**User Story for the Next Iteration: Partially Implemented System**

• **User Role**: Student, Faculty, and Alumni of University of Michigan

• **Goal**: Enhance the search functionality and recommendation system for the JSPR Techies blog.

• **Reason**: To provide users with improved search results and article recommendations based on their preferences and interests.

**Sub-Stories for the Next Iteration:**

1. **Search Initiation Enhancement**

**Size**: 3

**User Role**: Student, Faculty, and Alumni of University of Michigan

**Goal**: Allow users to enter search keywords.

**Reason**: To provide users with the ability to search for articles based on their interests. **Preconditions**:

Users should be logged into the JSPR-Techblog.

The search bar is accessible on the user interface.

**Triggers**:

The user accesses the JSPR-Techies blog.

**Postcondition**: Users should be able to initiate a search by clicking on the submit button. **Exceptions**:

If the user is not logged in, they will be prompted to log in before initiating a search. Technical issues affecting the search functionality may prevent users from initiating a search.

2. **Search Recommender Improvement**

**Size**: 3

**User Role**: Developer

**Goal**: Enhance the recommendation system to recommend the top 5 articles by matching entered keywords to an AI model.

**Reason**: To provide users with more accurate and relevant article recommendations. **Preconditions**:

Users should have entered keywords to initiate the search.

The AI model for recommendations is in place.

**Triggers**:

The user initiates a search or provides search keywords.

**Postcondition**: Improved article recommendations will be returned.

**Exceptions**:

If the AI model encounters errors or fails to provide recommendations, users may receive suboptimal or no recommendations.

3. **Search Visualization Enhancement**

**Size**: 3

**User Role**: Student, Faculty, and Alumni of University of Michigan

**Goal**: Display the top 5 articles recommended by the AI model on the recommended articles page.

**Reason**: To make it easier for users to access and explore recommended articles. **Preconditions**:

Search recommenders should have identified articles matching search keywords.

12

**Triggers**:

Users access the recommended articles page.

**Postcondition**: Users should be able to click on the required article from the list of articles. **Exceptions**:

If the recommended articles are not available or if there are technical issues in displaying them, users may not be able to access the recommended articles.

**Functionality at the End of this Iteration:**

At the end of this iteration, the system will have enhanced search functionality, improved article recommendations, and a better user experience. Users will be able to initiate searches, receive more accurate recommendations, and easily access the recommended articles. The system's search and recommendation capabilities will be refined to offer a more tailored and satisfying user experience.

**Backlog Status:**

• The selected user stories for this iteration are those related to improving search and recommendations. They have been prioritized for implementation.

• Remaining user stories in the backlog are not being implemented in this iteration. They will be considered for future iterations.

The overall objective is to make the JSPR-Techies blog more user-friendly and content-relevant, focusing on the search and recommendation aspects in this iteration.