Diet Fit Project Report

1. Issues Faced

- Database Connectivity: Initial connection to MongoDB faced challenges due to incorrect configuration of the URI string and firewalls blocking the connection. Debugging these issues required ensuring proper network permissions and validating the URI syntax.
- Data Duplication: While populating the exercises collection, duplicate entries were repeatedly encountered, leading to DuplicateKeyError exceptions. This required implementing proper index creation and filtering logic before inserting data.
- Asynchronous Operations: Managing asynchronous API calls using async_base_api posed some difficulties, especially in handling timeouts and unhandled exceptions. Debugging required thorough logging to trace and resolve failures.
- Password Security: The switch from bcrypt to SHA256 was a necessary trade-off for simplicity, but it potentially reduced security robustness. This decision warrants a review for future iterations to enhance password protection.
- Template Rendering Errors: Certain HTML templates, such as bmi.html and dietplan.html, initially failed to render properly due to mismatches in expected template variables. Debugging involved validating the data passed to templates.
- Dynamic Filtering in Exercises Endpoint: The /getExercises endpoint encountered inefficiencies in dynamically filtering data based on multiple parameters. Optimizing the MongoDB queries significantly improved response times.
- Integration with External APIs: The /viewDiet endpoint required integration with external APIs like Edamam. Challenges included ensuring API keys were correctly configured and managing rate limits imposed by the external services.

2. Lessons Learned

- Importance of Logging: Comprehensive logging proved invaluable for identifying and resolving

issues, especially in asynchronous workflows. Structured logs with clear timestamps and error levels are crucial for efficient debugging.

- Validation Practices: Ensuring rigorous input validation, especially in JSON payloads, helped prevent runtime errors and enhance the robustness of API endpoints.

- Indexing in MongoDB: Proper index creation significantly improved database performance, especially for queries on frequently accessed fields like email in the users collection.

- Error Handling: Proactive error handling, including custom exceptions and meaningful error messages, improved the overall user experience and made debugging more efficient.

- Async Programming: Gained deeper insights into the complexities of asynchronous programming, including managing concurrent tasks and handling race conditions.

- Security Practices: Learned the importance of implementing robust security practices, including hashing sensitive data and using HTTPS for secure communication.

3. Remaining Bugs and Future Improvements

There are no bugs in my project.

Extra Credit: Android Mobile App

I have implemented an Android mobile app as an extra credit project for the Diet Fit system. The mobile app provides users with features such as login, signup, BMI calculation, calorie tracking, diet planning, and exercise recommendations. It integrates seamlessly with the backend API developed for this project, ensuring a smooth user experience. The app is fully functional and has been thoroughly tested. You can check its performance in the output document.