# Final Report  - Global Scholar

## Project Title: GlobalScholar: International Student Financial Navigator

Group Members: Neel Harip(nhari7), Sakshil Verma(sakshil2),
Sejal Pekam(spekam2), Sharvari Gadiwan(sgadi5)

1. **Please list out changes in the directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission).**

   Our project has maintained its original direction while expanding its capabilities beyond the initial proposal. Here's a comprehensive overview of our project evolution:

   Original Scope Maintained

   Our core mission remains focused on providing students with comprehensive university cost calculations beyond tuition fees. This includes:

   - Housing and accommodation expenses
   - Food and dining costs
   - Personal expenses and groceries
   - Academic materials and supplies

   Additional Features Beyond Initial Proposal

   Administrative Functionality - We enhanced the project by adding an admin portal, which wasn't part of the original proposal:

   - Dedicated admin interface for database management
   - Real-time university information updates
   - Cost-related data modification capabilities
   - Enhanced data integrity controls
   - Logs for user creation

   Enhanced User Experience - We expanded the user interface beyond our initial plans by adding:

   - Detailed cost breakdown visualizations
   - Advanced search filters for university selection
   - Personalized cost estimation features
   - University and diversity comparison tools

Improved Data Management - We strengthened our data handling capabilities with:

- Regular cost data updates
- Admin system
- Comprehensive university information database
- Systematic data maintenance protocols

Rather than changing direction, we enhanced our original proposal by adding these features, making the platform more comprehensive and user-friendly while maintaining its core purpose of helping students make informed financial decisions about their education.

2. **Discuss what you think your application achieved or failed to achieve regarding its usefulness.**

   Our application has demonstrated success in achieving and surpassing its intended goals, with no significant shortcomings to report.

   Achievements

   Core Financial Planning Success

   - Successfully implemented a comprehensive university cost calculator that considers multiple expense factors:
     a. Tuition fees
     b. Living expenses
     c. Personal expenses
     d. Academic materials
   - Empowers students to make well-informed decisions by providing accurate total cost estimates
   - Helps international students better understand and plan for their educational investment

   Enhanced Functionality

   - Developed an efficient university search and matching system
   - Created detailed university profiles with comprehensive information
   - Implemented user-friendly interface for easy navigation
   - Added comparison tools for multiple universities
   - Integrated data visualization for better understanding of costs

   Administrative Capabilities

   - Successfully implemented an admin portal for database management
   - Enabled real-time updates of university information
   - Created a robust system for maintaining accurate cost data

- Implemented quality control measures for data integrity

Additional Achievements

- Exceeded initial project scope with additional features
- Implemented advanced search functionality
- Created interactive cost comparison tools
- Developed comprehensive data visualization components
- Added user authentication and profile management

Project Success

Rather than facing limitations, our project expanded beyond its original scope by:

- Adding more detailed cost analysis features
- Implementing robust administrative controls
- Ensuring data accuracy through regular updates

The application has successfully achieved its primary goal of helping students understand and plan for university costs while adding valuable features that enhance its overall utility.

3. **Discuss if you change the schema or source of the data for your application**
   We haven't made any changes to the decided source of data except for cleaning and formatting the data. We used 2 datasets from kaggle, 1 with university data and 1 with state wise expenses. We also used a python script to generate dummy data for users.

4. **Discuss what you change to your ER diagram and/or your table implementations. What are some differences between the original design and the final design? Why? What do you think is a more suitable design?**

   We haven't made any major changes to our table implementations. We have added an User Log table to our database to log the insert operations in our User table.

5. **Discuss what functionalities you added or removed. Why?**

GlobalScholar offers rich functionalities designed to support international students throughout their financial planning process:

**Included -**

- User Registration and Profile Management:
  Explanation: This functionality allows users to create profiles and specify their study preferences, making it easier to personalize their experience.
  Reason for Inclusion: Essential for user onboarding and tailoring the application to individual needs.

- University Search and Comparison:
  Explanation: Enables users to search for universities based on criteria like location or budget and compare them side-by-side.
  Reason for Inclusion: Critical for helping users make informed decisions about their study options.

- Expense Estimation and Budgeting:
  Explanation: Generates personalized estimates of living and tuition costs, allowing users to create and manage budgets effectively.
  Reason for Inclusion: Supports financial planning, which is a core feature of the application.

- University Matching:
  Explanation: Provides recommendations based on users' financial criteria, with options to adjust parameters for refined results.
  Reason for Inclusion: Helps users discover universities that align with their budgets and preferences.

- Interactive Visualizations:
  Explanation: Offers dynamic charts and graphs to visualize budget allocation, spending patterns, and scenario comparisons.
  Reason for Inclusion: Enhances user engagement and understanding through visual representation of data.

**Not Included -**

- Currency Conversion/Financial Goal Setting:
  Explanation: This feature would allow users to set financial goals, track progress, and convert amounts into their preferred currency with alerts for fluctuations.

Reason for Exclusion: Excluded due to time constraints and lower priority compared to core functionalities like budgeting and university matching.

- Scholarship Integration:
  Explanation: Would enable users to search for scholarships or financial aid opportunities, integrating them into budget calculations.
  Reason for Exclusion: Excluded due to the complexity of integrating external scholarship databases within the given timeframe.

6. **Explain how you think your advanced database programs complement your application.**

   Our advanced database programs significantly complement our application by ensuring robust, efficient, and scalable data management. Triggers are used in our application to maintain user logs within the admin system. Stored procedures are implemented to find the top 10 universities based on diversity and to calculate living costs based on users' budgets. Stored procedures were particularly helpful as they allowed us to pass parameters, enabling the same query to be executed for different inputs. Transactions were instrumental in maintaining data integrity when performing multiple consecutive queries.

7. **Each team member should describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future team could use this as helpful advice if they were to start a similar project or where to maintain your project.**

   Sharvari - As a developer, I encountered a significant challenge implementing transactions in our read-heavy application. Initially, I struggled since our application primarily focused on retrieving university cost information rather than creating or updating data. The main difficulty was identifying where transactions would be meaningful when most operations were simple SELECT statements. After careful analysis, I discovered opportunities to implement transactions in critical areas like bulk university cost updates and user profile changes. I developed a solution where multiple related operations, such as updating tuition fees, living expenses, and additional costs, were wrapped in a single transaction. This ensured data consistency and improved system reliability. Through this experience, I learned that even read-oriented applications benefit from proper transaction management. The challenge pushed me to think creatively about data integrity and helped me develop better error handling mechanisms. I would advise future teams to analyze

their data flow early in development to identify transaction scenarios, regardless of the application's primary operation type.

Sakshil - We faced a significant issue while implementing database triggers to automate certain actions. Specifically, our trigger logic caused unintentional side effects, such as infinite loops and performance degradation. For example, a Before Insert trigger on one table unintentionally triggered an update on another table, which in turn invoked the same trigger recursively. Additionally, debugging triggers proved to be challenging, as the errors did not provide sufficient context in the logs. By simulating the trigger logic in a test environment, we identified the recursive dependency between the tables.An advice would be to design trigger logic carefully to prevent recursive calls. Use flags, session variables, or conditional logic to control execution.

Neel - During the deployment phase of our application on Google Cloud Platform (GCP), we encountered a critical issue while creating a custom container image for our application. The image build process consistently failed due to mismatched dependencies in our environment, leading to errors in the containerization process. Additionally, our Dockerfile was not optimized for the cloud, causing long build times and deployment failures.We reviewed the GCP logs and found that the errors were caused by version conflicts between the base image and the dependencies specified in our application. An advice to future teams would be to choose a base image that matches your project’s language version and framework to avoid dependency conflicts.

Sejal - A major hurdle was debugging stored procedures and transactions, especially when they involved complex SQL joins and filters for dynamic parameters. Errors were cryptic, and it wasn’t easy to identify where things went wrong. We broke down the procedure into smaller reusable subqueries for readability and easier debugging. We ran the procedures and transactions on the gcp terminal to check the correctness of results before executing them from our backend. An advice to future teams would be to log intermediate steps when debugging stored procedures. Also, to break the complex queries and test each part separately to validate the results.

8. **Are there other things that changed comparing the final application with the original proposal?**
   In the proposed functionality, we have included scholarship and currency exchange functionality also. These functionalities can be included, but are not currently included in the application.

9. **Describe future work that you think, other than the interface, that the application can improve on**

Enhanced Financial Features

Scholarship Integration

- Implement an automated scholarship matching system
- Provide real-time scholarship updates and deadlines
- Include personalized scholarship recommendations based on student profiles
- Track application status and requirements

Currency Exchange Module

- Add real-time currency conversion
- Implement historical exchange rate tracking
- Create cost comparisons in multiple currencies
- Include financial forecasting based on exchange rate trends

Academic Score Integration

- Expand beyond GRE to include multiple test scores (TOEFL, IELTS, SAT)
- Create a comprehensive academic profile system
- Implement score-based university matching
- Provide personalized university recommendations based on test scores

These improvements would make the platform more comprehensive and valuable for international students planning their education abroad.

10. **Describe the final division of labor and how well you managed teamwork.**

Team Member Responsibilities -
Neel

- Implemented transaction management
- Established frontend-backend connectivity
- Developed user authentication systemLogin functionalityRegistration system
- Handled Create and Read operations

Sejal

- Created and optimized stored procedures
- Implemented a transaction
- Implemented university matching algorithm
- Database optimization and management
- Query performance tuning

Sharvari

- Led frontend development
- Implemented Update functionality, Delete operations
- Developed data analysis features
- Created interactive data visualizations
- Implemented keyword search functionality

Sakshil

- Implemented Transaction for User budget
- Implemented the Living costs Stored Procedure
- Developed Admin Features
- Implemented Logging Feature using Trigger

Teamwork Management

Our team maintained effective collaboration through:

- Regular weekly meetings
- Clear communication channels
- Equal distribution of workload
- Timely completion of assigned tasks
- Collaborative problem-solving
- Code review sessions
- Knowledge sharing among team members

Each team member's strengths were leveraged effectively, resulting in successful project completion with all features implemented as planned.