Vivek Reddy Bhimavarapu

- Designed and wrote the the Quiz&Flashcard schema
- Edited teammates sql's in the repository primarily in the load folder for seeding and building without errors. Added comments of what I did on the top of the file.
- Wrote the master script file for creating the study buddy databases that uses everyone's schemas.
- Changed directories of multiple files in the repository for proper build, and organization
- Wrote the query optimization analysis for my schema
- Web scrapped a geeks for geeks quiz to insert into quiz tables.
- Send the scrapped data of quizzes to Rise to clean it.
- Recorded the video for my part.
- Added the database execution output from the video presentation.

Rise Akizaki

- Wrote the SQL for the User_Management section.
  - Created all relevant tables (Users, Colleges, Courses, Majors).
  - Tested functionality of the SQL by using test queries (indexing, searching, etc).
  - Tested foreign key relationships between all the tables.
- Created scripts to clean the data for courses, and quizzes.
  - Created new fields to match the table format.
  - Added missing information required for each field.
  - Altered values to match their respective fields.
  - Outlined the process for creating the data cleaning scripts.
- Created an additional cleaning script for colleges to make the data match the table format.
  - Removed unnecessary fields
- Inserted scraped data for the courses, and colleges tables.

Sarah Kayembe

- Scraped Colleges & Universities
  - Identified a reliable public source (Wikipedia-accredited institutions).
  - Extracted institution names, states, and categories.
  - Cleaned inconsistencies, and removed duplicates
  - Saved as CSV for import (us_colleges.csv).

- Scraped List of Majors / Academic Programs
  - Located major listings from valid sources.

    - Extracted major names.
    - Cleaned and standardized capitalization.
    - Removed duplicates and empty entries.

- Scraped Courses for USM
    - Iterated through the paginated course catalog.
    - Collected course codes, titles, and sources.
    - Validated course structure.
    - Exported to CSV.

- Scraped Study Resources (Learning Materials)
    - Collected online study resources linked to course topics.
    - Removed irrelevant results, duplicates, and broken URLs.
    - Ensured resources had valid titles and source links.
    - Normalized data and aligned each resource with a course.

- Attempted Scraping Quizzes & Flashcards
    - Built a working scraper that collected practice questions and answers.
    - Ensured schema alignment (Quiz → Question → Answer).
    - Later removed from final deliverables so teammates could handle it.

- General Cleaning
    - Normalized text (trimmed spaces, removed HTML noise, fixed unicode).
    - Dropped duplicates using Pandas.
    - Ensured all mandatory fields were present.
    - Verified foreign key references existed before inserting.

- Validation Rules Implemented
    - Checked regex formatting for titles (e.g., monthly challenge titles).
    - Ensured date rules: *end_date > start_date focus sessions had valid time ranges*
    - Verified numeric ranges for:
        - durations
        - streak values
        - reward thresholds

- Ensured Foreign Key Integrity
    - Built lookup dictionaries (e.g., user_id → stats, topic_id → names).
    - Checked that every row matched an existing FK entry BEFORE insertion.
    - Logged anomalies (none appeared after cleaning).

- Inserted All Scraped Data Into MySQL to test and perform
    - Inserted colleges, majors, courses, and resources.
    - Inserted mood levels, item types, task categories, and topic types.
    - Inserted all focus items with categories and rarity levels.

  - Inserted daily focus logs with:
    - sessions
    - total focus time
    - topics
    - JSON fields (validated)
    - always-valid item_earned_id (tiered reward system)

- Created Additional Gamification Tables With Synthetic Data
  - userfocusitems

  - usercitylayout
  - monthlychallenges
    userchallengeprogress
  - moodtracking
  - leaderboardstats
  - studyStats (fully computed using SQL + Python)

- Built Synthetic Data for Testing
  - Generated:
    - 100+ daily focus logs
    - 100 monthly challenges
    - mood logs
    - challenge progress
    - leaderboard entries

  - Ensured no NULLs in required fields.
  - Verified JSON validity before insertion.

- Completely wrote the SQL script for the study_management page and normalized it.
  - Broke ENUMs into reference tables:
    - topic_types
    - task_categories
    - item_types
    - mood_levels

- Query Optimization & Indexing
  - idx_task_user_status_due
  - idx_task_course_status
  - idx_timer_host_start
  - idx_timer_host_end_duration
  - idx_participant_user
  - uq_user_date (unique composite index)

  - idx_focus_type
  - idx_challenge_reward
  - idx_progress_challenge
  - idx_city_user
  - idx_stats_topic
  - idx_leader_user

- Ran Query Performance Analysis
- Compared Performance With & Without Indexes
- Stored Procedures & Backend Logic (Study Management System): I implemented several key stored procedures that support the Study Management system's core functionality, including:
  - Procedures for adding, updating, and retrieving tasks.
  - Procedures for inserting timer sessions and calculating session durations.
  - Procedures for updating user study statistics (total minutes, streaks, averages).
  - Leaderboard update procedures and challenge-progress procedures.
  - Additional Deliverables
    - Saved all data as CSV for reproducibility.
  - Produced a video demonstration of the scraped data and code.

Jacob Craig

- Designed and implemented all tables related to the Study Group & Collaboration section
- Created Test data for the Study_Groups & Collaboration section
- Built a Python data cleaning script that takes a CSV file for course resources and:
  - Removes duplicates
  - Cleans up whitespace and inconsistent patterns
  - Normalizes title formats
  - Outputs a CSV compatible with MySQL
  - Generates SQL Code for the resource table
    - Looking to improve upon this by eventually automatically inserting from the program into the database
- Created a full query workload with procedures and created indexes for the Study Groups & Collaboration section of the database
- Created automated triggers, procedures, and constraints for the Study Groups & Collaboration section of the database
  - Created a group_summary table to help with optimization
  - Designed three automated triggers to:
    - Increment member count on join
    - Decrement on removal
    - Update the latest session on new session creation
- Recorded working Resource Cleaner Program and working procedures