As a team, in a document (called EECS2311-SectionName-Team#-TakeHomeAssignment.pdf) list the fully implemented and unit/integration tested user stories in your project up to now that have GUI as well.

1. Financial Transaction Logging (Joshua Khalifa)

User Story: As a user, I want to record my financial transactions (income and expenses) so that I can track where my money is going.

Implementation:
- TransactionLogUI and TransactionController provide complete transaction management
- Users can categorize transactions, add amounts, and set dates
- Transaction history view with filtering options

Testing:
- Unit tests validate transaction data stored correctly in database
- Integration tests confirm transactions appear in views and reports

2. Budget Creation and Monitoring (Joshua Khalifa)

User Story: As a user, I want to set weekly and monthly budget limits so that I can control my spending habits.

Implementation:
- BudgetController provides budget management functionality
- Users receive notifications when approaching or exceeding limits
- Visual indicators of budget status

Testing:
- Unit tests validate budget calculations
- Integration tests for notification triggers

3. Financial Goal Setting and Tracking (Joshua Khalifa)

User Story: As a user, I want to set financial goals with deadlines so that I can save for specific purposes.

Implementation:
- GoalsUI and GoalController enable creating and tracking goals
- Progress visualization with completion percentage
- Goal completion events award XP

Testing:
- Unit tests for goal progress calculation

- Integration tests for goal completion effects

4. Automatic Quest Completion System (Joshua)

User Story: As a user, I want to automatically earn rewards when I complete financial tasks so that I'm motivated to maintain good financial habits.

Implementation:
- QuestsUI provides a quest board interface
- QuestController automatically checks and completes quests based on user activity
- Timer-based refresh for real-time updates

Testing:
- Unit tests for quest completion conditions
- Integration tests across multiple controllers

5. Transaction Categorization System (Jiwon)

User Story: As a user, I want to categorize my transactions into predefined and custom categories so that I can analyze my spending patterns more effectively.

Implementation:
- CategoryManager for managing transaction categories
- Category selection UI in transaction forms
- Color-coded category visualization
- Smart category suggestions based on transaction descriptions

Testing:
- Unit tests for category creation and assignment
- Integration tests with analytics system
- Tests for category persistence across sessions

6. Financial Analytics Dashboard (Peyton)

User Story: As a user, I want to see visual analytics of my spending patterns so that I can better understand my financial behavior.

Implementation:
- AnalyticsUI and AnalyticsController with charts and graphs
- Spending breakdown by category
- Income vs. expense comparisons

Testing:
- Unit tests for data aggregation algorithms

- Integration tests for chart generation

7. User Authentication System (Khalifa)

User Story: As a user, I want to securely log in to the application so that my financial data remains private.

Implementation:
- LoginScreen and RegisterScreen interfaces
- UserController for authentication logic
- Password security measures

Testing:
- Unit tests for validation and authentication
- Integration tests for session management

8. Transactions view history (Andy)

User Story: As a user, I want to view and filter all of my transaction data.

Implementation:
-Transaction history with database

Testing:
- Unit tests for transaction
- Connection with databse

9. Financial Literacy Quiz System (Andy)

User Story: As a user, I want to test my financial knowledge through quizzes so that I can improve my financial literacy while earning rewards.

Implementation:
- QuizUI provides interactive quiz interface
- QuizController manages questions and scoring
- XP rewards for correct answers

Testing:
- Unit tests for quiz scoring
- Integration tests with XP system

10. CSV Import for Financial Transactions (Jiwon)

User Story: As a user, I want to import my bank statement CSV files so that I can quickly add multiple transactions without manual entry.

Implementation:
- CSV import functionality in TransactionController
- File selection dialog for choosing CSV files
- Column mapping interface for flexible import formats
- Transaction preview before final import

Testing:
- Unit tests for CSV parsing logic with various formats
- Integration tests with transaction database to ensure proper storage
- Error handling tests for malformed CSV files

2. In the document, explicitly identify who are the main developers of each user story.

1. Joshua Khalifa
2. Joshua Khalifa
3. Joshua Khalifa
4. Joshua
5. Jiwon
6. Peyton
7. Khalifa
8. Andy
9. Andy
10. Jiwon

3. Evenly distribute all stories to team members such that nobody is assigned to a user story that they are heavily involved in its implementation.

Andy 2 4
Joshua 5 10
Jiwon 1 3
Khalifa 6 9
Peyton 7 8

4. Write down who is now responsible to test which stories and place this document in your team's GitHub.

Andy 2 4
Joshua 5 10
Jiwon 1 3
Khalifa 6 9