Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Faculty Kit

The objective of this project is to automate the interest calculation process for a retail bank. Customers deposit money in accounts and receive interest based on a pre-determined rate. The system calculates daily interest for accounts with positive balances and stores it cumulatively. At the end of each month, this cumulative interest is credited to the customer's account.

Evaluation Strategy/Tips for the different milestones of the project

Objective
The Faculty Kit provides an evaluation strategy for assessing the progress and quality of the project at various milestones. It includes guidelines and tips for assessors along with useful resources and documents for evaluation.

Requirements Specification
Key Evaluation Points:

- Clarity and completeness of the functional and non-functional requirements.
- Validity and feasibility of assumptions made.
- Team-wide understanding of the requirements.
- Presentation and documentation quality (organized and professionally formatted).

Technology Familiarization
Mode: Team presentation + Transcation + InterestCalculation.

Technologies to be covered:

- React, TypeScript (Frontend)
- Node.js with Express.js (Backend)
- MongoDB (Database)
- Render (Frontend Deployment)
- Render (Cloud-hosted MongoDB)

Evaluation Criteria:

- Clear understanding of each technology's role.
- Ability to explain architecture and tech stack choices.

Database Creation
Evaluation Parameters:

- Normalized, clear schema with well-defined document structures and logical relationships between collections (e.g., Customers, Transactions, InterestLogs).
- Proper use of datatypes and constraints.
- Proper use of datatypes and constraints in MongoDB (e.g., using appropriate types for Date, Number, String, and referencing ObjectIDs where needed).
- Inclusion of backup/recovery strategies and indexing where necessary.
- Documentation of relationships (one-to-many, many-to-many).

High-Level and Detailed Design
Evaluation Points:

- The design includes all key features: account management, transaction recording (deposit/withdraw), daily interest calculation, monthly interest crediting, and admin controls.
- Also covers dynamic balance updates and historical interest tracking per user.
- Clarity of flowcharts or pseudocode (e.g., daily interest = (balance × rate) / 365).
- Alternatives or scalability considerations discussed.
- Error handling for DB connection failures, missed cron jobs, and transaction rollbacks.

Front-end implementation

Demo Requirement: Fully working front-end without backend integration

Evaluation Criteria:

- Clean and user-friendly UI using HTML/CSS.
- Proper layout and accessibility practices.
- Responsiveness across devices.
- Intuitive Navigation and Meaningful Content Hierarchy (including account, interest, and transaction sections)
- Presence of Tooltips, Modals, and Basic Validations (JS) (for displaying interest details and handling form actions like deposit/withdrawal).

Integrating the front-end with the database
Evaluation Points:

- Data Management: Efficient management of account records including adding, viewing, modifying, and removing entries.
- Stability: Smooth operation without crashes, with meaningful error messages on failures.
- Demonstration of:
  - Login/logout session handling.
  - New user creation and deletion.
  - Record addition, editing, searching, and deletion.
- Deployment: Frontend deployed on Render and database also is on Render.

Test-plan review
Evaluation Criteria:

- Coverage of All Functional Requirements: Ensure all key functionalities are tested.
- Test Case Execution Descriptions: Provide clear steps to execute each test case.
- Inclusion of Negative Test Cases: Test for invalid inputs and other edge cases.
- Exception testing (e.g., DB disconnection, Render deployment issues).
- Use of tools like Postman for API testing is a plus.

Final review
Deliverables:

- Final working project (deployed).

- Project Report.
- All documentation from prior stages.

Evaluation Points:

- Completeness and functionality of the Bank system.
- Smooth demo across modules (login, admin control, User management, Interest Management).
- Quality of code (structure, readability, comments).
- Project Report includes architecture, challenges, screenshots, future scope.
- Deployment links and accessibility.

Documents/References that may aid the process of evaluation

- **HTML/CSS/JS**: https://www.w3schools.com/
- **Node.js**: https://nodejs.org/doc
- **MongoDB**: https://www.mongodb.com/docs/