TechTrek 2023 Challenge Statement

3rd December 2022

Introduction

With the expansion of DBS into new markets, we must cater our banking services to the needs of new users in our new markets. A core foundation of all digital banking services is a robust online banking web application.

Challenge Statement

Your task is to create a simplified full-stack digiBank web application that allows customers to access and schedule transactions with their bank account. This allows our customers to get access to banking services wherever, whenever, enabling them to Live more, Bank less. This **must** be a **web application**.

*For each module, there will be a Frontend task and an accompanying Backend task as illustrated by the table below.

Module		Basic Requirements (Frontend)	Basic Requirements (Backend)
Login	**[1]**	User must be able to login	Server must be able to authenticate a user's identity
Dashboard	**[2]**	Display customer's Bank Account type and balances	Return a list of account information of a user from the ***Bank Account*** table
	[3]	Users must be able to view all their scheduled transactions	Return a list of transaction details of a user from the ***Scheduled Transactions*** table
Transactions	**[4]**	Schedule new transactions	Insert transactions created from frontend into ***Scheduled Transactions*** table
	[5]	Delete scheduled transactions	Delete scheduled transactions from the ***Scheduled Transactions*** table
	[6]	Provide, change, or remove their address and email	Return and Update from list of user details from the ***User*** table

Basic Application Requirements (Frontend):

- You must render a login page
  - User must be able to login **[1]**.
- You must render a dashboard
  - Display customer's Bank Account type and balances **[2]**.
  - Users must be able to view all their scheduled transactions **[3]**.
- Users must be able to:
  - Schedule new transactions **[4]**
    - Enter recipient's bank account id.
    - Enter or select a date value.
    - Enter a transaction amount.
    - Optionally enter a comment for the recipient.
  - Delete an existing scheduled transaction **[5]**.
  - Provide, change, or remove their address and email. **[6]**.

Basic Application Requirements (Backend):

- You must set up a valid authentication API
  - Server must be able to authenticate a user's identity **[1]**.
- You must set up the respective API with the following functionalities:
  - Return a list of account information of a user from the ***Bank Account*** table **[2]**.
  - Return a list of transaction details of a user from the ***Scheduled Transactions*** table **[3]**.
  - Insert transactions created from frontend into ***Scheduled Transactions*** table **[4]**.
  - Delete a future transaction entry from the ***Scheduled Transactions*** table **[5]**.
  - Return and Update from list of user details from the ***User*** table **[6]**.

Basic Application Requirements (Integrate):

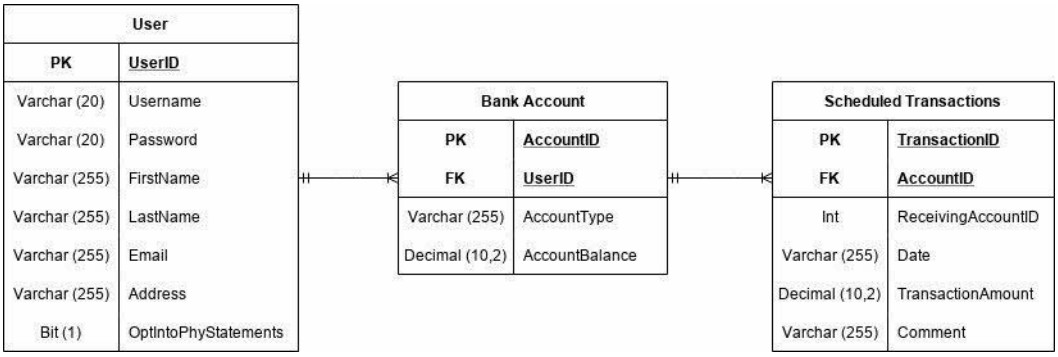
- Integration is a crucial requirement for this hackathon. The front end and back end should be ***integrated*** seamlessly.

Data provided

You will be given:

- Data in *JSON* and *SQL* format
  - The entries provided are not exhaustive and you can add more to suit the needs of your application
  - Entity Relationship Diagram (ERD). The following ER diagram is provided as a reference. We have provided datasets for ***User, Bank Account,*** and ***Scheduled Transactions*** entities.
  - The ***Bank Account*** and ***Scheduled Transaction*** tables provided can be used to show how the CRUD is being implemented. These tables are part of the requirements **[2-6]**. You are free to add new attributes to match any new features you wish to introduce; teams **MAY** make changes so long as tasks requirements are met.

ERD Diagram



Database Example

User Table

UserID	1
Username	ExecutiveDBS
Password	DBSBestBank2022
FirstName	Tom
LastName	Lim
Email	tomlim@easymail.com
Address	Block 123 Serangoon Garden #10-129
OptIntoPhyStatements	0

Bank Account Table

Id	UserID	Type	Balance
621156213	1	Saving	70200.71

Scheduled Transaction Table

ID	AccountID	ReceivingAccountID	Date	Amount	Comment
1	621156213	339657462	2022-11-08T04:00:00.000Z	500.00	Monthly Pocket Money

Extension Modules

The following modules are extensions and are fully optional. You should tackle these extensions individually and showcase your knowledge in the relevant pillars. To recap, the pillars are:

1) Application Development and Support
2) DevOps and Site-Reliability Engineering
3) Data Engineering & Artificial Intelligence/Machine Learning
4) ICT Infrastructure

You will showcase your work in these modules together with the main challenge during the presentation segment of TechTrek. Do remember to highlight the pillar you are showcasing your skills in.

Do note that you are expected to finish the main challenge before working on the extension modules.

Pillars	Modules	Requirements	
Application Development and Support		Front-end	Back-end
	DEV 1	Auto log out of account after 5 minutes of inactivity	JSON Web Tokens for authentication
	DEV 2	Display a phishing warning upon every N^{th} login	Keep track of number of logins from a particular user since last phishing warning was displayed
	DEV 3	Other features	
DevOps and Site-Reliability Engineering	**SRE 1**	Write Unit Testing	
	SRE 2	Apply SRE Principles	
Data Engineering & Artificial Intelligence/Machine Learning	**AI 1**	Data treatment and cleaning	
	AI 2	Develop a Logistic Regression Model	
ICT Infrastructure	**ICT 1**	Explore methods to apply Dockerization to your application	
	ICT 2	Utilize AWS/GCP to deploy application to cloud	

DBS Live more, Bank less

Application Development and Support Modules

[DEV 1]

Authentication plays an important role in the security of a digital bank. For this module, you will work in a pair (Frontend and Backend) to develop a working authentication system with an auto log out functionality.

Requirements (Frontend):

- Work with Backend to provide JSON Web Token functionality for authentication
- Auto log out of account after 5 minutes of inactivity
- JSON Web Token should also be erased during log out

Requirements (Backend):

- Provide JSON Web Token functionality for authentication
- Provide API end points for Frontend
- Ensure JSON Web Token is erased upon user log out or time out

[DEV 2]

From time-to-time, we are required to disseminate important messages to our customers via the digiBank platform. For this module, you will work in a pair (Frontend and Backend) to display a friendly phishing reminder message to remind our customers about the risks of phishing and simple preventive measures that our customers can be aware of. This message should be displayed upon successful login.

Requirements (Frontend):

- Consider interrupting vs. non-interrupting user experience (UX) and what's best for this use-case.
    - You are free to determine the mode of displaying this message.
- Only display the message on every N^{th} login for each user to minimize friction and prevent users from becoming desensitized to service messages.
    - You are free to determine the ***N*** value that is most appropriate.

Requirements (Backend):

- For each user, extend the ***User*** table to support the frontend in implementing this feature.

[DEV 3]

This module is free form and allows you to decide what features or enhancement to existing features could benefit our user.

Requirements:

- Program an additional feature to your application that would help the user's journey using your application
- Explain how this feature can benefit the user in relation to the user's journey

DevOps and Site-Reliability Engineering Modules

[SRE 1]

To ensure the application functions as intended, it is important to write unit tests to check for software failures. For this module, you will be writing unit tests for your application.

Requirements:

- Write sufficient unit testing for your application
- Ensure that there is at least 60% coverage under statements, branch, functions, and lines

[SRE 2]

To build a functional application, it is important to design and promote a service management strategy that works for your developed product. For this module, you will utilize Site-Reliability Engineering (SRE) principles for your application.

Requirements:

- Identify Service Level Objectives and Indicators (with explanation of choice)
- Develop risk acceptance and mitigation plan

Data Engineering & Artificial Intelligence/Machine Learning Modules

[AI 1]

As the world becomes more digitalized, data is generated rapidly across multiple platforms. In order to make use of these data productively, data treatment is necessary before they can be fed into models to create artificial intelligence. In this module, you will be tasked to perform data treatment on a similar dataset separate from the main challenge. The dataset will be given to you in csv format. It is recommended that you perform this task in Python or Excel.

ClientProfile_Dataset:

The dataset provided to you will have the following format:

- **FirstName** (String)
- **LastName** (String)
- **Address** (String)
    - This field structure is consistent, and it consists of a block number, street name (can contain two or three words), unit number and postal code.
- **MonthlyIncome** (int)
- **AmountInStocks** (String)
    - This field will take on either a "High" or "Low" value.
- **AmountInBonds** (String)
    - This field will take on either a "High" or "Low" value.
- **HasRelationshipManager** (String)
    - This field will take on either a "Yes" or "No" value.

Note: A client with "High" in ***"AmountInStocks"*** will have "Low" in ***"AmountInBonds"*** and vice versa.

DBS Live more, Bank less

Requirements:

- Fill in the missing data in ***"AmountInStocks"***, ***"AmountInBonds"***, ***"HasRelationshipManager"*** and explain your method for doing so. Explain your considerations if any.
- Extract the street name and postal code from the ***"Address"*** column into two new columns ***"StreetName"*** and ***"PostalCode"*** respectively. Change the data type of ***"PostalCode"*** to int64.
- Add in a new variable ***"HighNetWorthIndividual"*** which should take on either a "Yes" or "No" value. A client is deemed a high net worth individual if his income is above $8000.
- Add in two new rows of data as shown below.

FirstName	LastName	Address	MonthlyIncome	AmountInStocks	AmountInBonds	HasRelationshipManager	Postal Code	StreetName	HighNetWorthIndividual
Tom	Henderson	Block 256 Downtown Road #27-2743 Singapore 256376	3356	Low	High	No	256376	Downtown Road	No
Grace	Maple	Block 624 Marymount Street #35-9234 Singapore 624298	7824	High	Low	Yes	624298	Marymount Street	No

[AI 2]

With the huge amount of data available, DBS has been utilizing artificial intelligence/machine learning to solve problems. However, there have been many problems in creating a good model and evaluating its effectiveness. In this module, you are tasked to create a binary logistic regression model and answer the following questions. The dataset will be given to you in csv format and it will be ready for use. It is recommended that you perform this machine learning task in Python.

ClientScore_Dataset:

The dataset provided to you will have the following format:

- **Score 1** (int)
- **Score 2** (int)
- **CreditWorthy** (int)
    - This field will take on a value of either 1 or 0.

Note: A value of 1 in **"CreditWorthy"** means that the client is credit worthy and vice versa.

Requirements:

- Create a binary logistic regression model and evaluate its performance.
    - Checking of the distribution of the training and testing data is not required.
    - For the ***"penalty"*** parameter, set it to **"none."**
- **[Optional]** Show the boundary of the logistic regression in a scatter plot.
- Explain your approach in dealing with the problem of an imbalanced dataset.
- Explain your approach in determining the evaluation matrix to be used to evaluate the performance of a model.
- Explain the bias-variance trade-off.

ICT Infrastructure Modules

[ICT 1]

With increased interest in cloud technology, Dockerization has developed to ease the burden between application development and cloud deployment. For this module, you will be looking into the application of Docker technology for your application.

Requirements:

- Explain how you would go about applying Dockerization for your application during deployment
- Showcase the expected benefits Dockerization can provide for the application or development team

[ICT 2]

Cloud services have seen an increased interest over the past few years, with more and more developers using cloud technology to deploy their applications. For this module, you will be looking into deploying your application through Amazon Web Services (AWS) or Google Cloud Platform (GCP).

Requirements:

- Choose either AWS or GCP to deploy your application
- Explain the benefits of your chosen cloud platform, and how it can benefit the company