DBS Live more, Bank less

TechTrek 2025
Challenge Statement

11th January 2025

Introduction

Climate change is one of the biggest challenges facing our planet today. One of the key strategies to combat climate change is reducing greenhouse gas emissions, particularly carbon dioxide (CO_2). Carbon credits are a market-based solution to help businesses offset their carbon emissions by investing in projects that reduce or capture CO_2, such as renewable energy or reforestation projects.

As the demand for carbon credits grows, it becomes essential to create an efficient, transparent, and user-friendly platform that enables businesses to trade carbon credits seamlessly. This platform should foster trust, ensure fair pricing, and help users track and manage their carbon offsets effectively.

Challenge Statement

Your task is to design and develop a carbon credit trading platform that allows users, such as companies, to trade carbon credits. The application should offer essential features such as viewing their available carbon credits, purchasing additional carbon credits from other companies, and selling carbon credits to other companies. This must be a **web application**.

All the requirements included below are meant to be a guideline to guide your team on how to tackle this challenge statement. You should treat this challenge like an actual work project, and not as a test. Do communicate actively with your assessors for any other additional areas to consider.

*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]	Users must be able to login	Server must be able to authenticate a user's identity using JSON Web Tokens for authentication
Landing Page	[2]	Display company's outstanding balances for: • Carbon Credits • Cash Balances	Return user's balance details from ***companyAccount*** table: *companyName, carbonBalance and cashBalance.*
	[3]	Display all outstanding requests made by your company with the following details: • Request Date • Company Name • Carbon Price (SGD/Tonnes) • Carbon Quantity • Requesting Reason • Request Type (Buy/Sell)	Return a list of outstanding requests from the ***outstandingRequest*** and ***companyAccount*** tables: *companyName, requestDate, carbonUnitPrice, carbonQuantity, requestReason and RequestType.*

	[4]	Insert/Edit/Delete requests made by your company with the following details: - Request Date - Company Name - Carbon Price (SGD/Tonnes) - Carbon Quantity - Requesting Reason - Request Type (Buy/Sell)	Able to create new requests, edit the request and delete the request from ***outstandingRequest*** table. Creating requests should also update the ***requestReceived*** table.
Requests Received Page	[5]	Display all outstanding requests made from other companies with the following details: - Request Date - Requestor Company Name - Carbon Price (SGD/Tonnes) - Carbon Quantity - Requesting Reason - Request Type (Buy/Sell)	Return a list of outstanding requests from the ***outstandingRequest*** and ***companyAccount*** tables: *requestorCompanyName, requestDate, carbonUnitPrice, carbonQuantity, requestReason and RequestType.*
	[6]	Create a button to Accept / Reject as well as checkbox to Bulk Accept / Reject requests made from other companies.	Update the database for ***outstandingRequest*** table to indicate accept / reject status. Updates the corresponding account balances in ***companyAccount*** table.
	[7]	Functionality upon entering the page to alert users via pop-up message for overdue requests (defined as 7 days after request date).	Returns and updates alerts as well as its view status stored within the ***requestReceived*** Table.

Bonus Challenges

Creating a Dashboard / Visualisation(s) to provide insights to companies:

Dashboard	[B1]	Display Data Visualisation tables or charts	Come up with simple visualisations deemed appropriate. E.g. Time series of requests in a particular month, time series of settlement prices, charts of request approval rates.
	[B2]	Display useful insights from data	Provide insights from the data visualized and corresponding recommendations for the company. You may incorporate external datasets for real carbon transactions.

Creating a Multi-Layer Approval Workflows for requests:

Multi-Layer Approval Workflow	[B3]	Requires for multiple approvers to approve a single request before it can be processed.	Modify the workflow and tables to require more than one approver before the request is set to approved and is processed.

Basic Application Requirements (Integrate):

- Integration is a crucial requirement for this hackathon. The frontend and backend components should be **integrated** seamlessly.

Basic Application Requirements (Frontend):

- You must render a login page
    - User must be able to login [1]
- You must render a landing page
    - Display user's current balances [2]
    - Display user's existing requests [3]
        - Company Name
        - Request Type
        - Carbon Price
        - Carbon Quantity
        - Requesting Reason
        - Request Date Timestamp
- User must be able to:
    - Create new Buy/Sell requests [4]
    - Edit existing requests made by their companies [4]
        - Edit one or more Buy/Sell requests from the landing page
        - User should not be able to edit requests of completed requests
    - Remove (Delete) pending requests made by their companies [4]
        - Remove one or more existing Buy/Sell requests
    - View requests made from other companies to them [5]
    - Accept/Reject requests made from other companies [6]
    - Be alerted of overdue requests [7]

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 user's current balances [2]
    - Display user's outstanding requests from the ***outstandingRequests*** table [3]
    - Insert requests provided by backend into the ***outstandingRequests*** and ***requestReceived*** table [4]
    - Edit outstanding requests made by user's company in the ***outstandingRequests*** table and ensure it is updated for each change [4]
    - Delete requests from ***outstandingRequests*** table [4]
    - View requests made from other companies from the ***outstandingRequests*** table [5]
    - Accept/Reject requests made from other companies and update the ***outstandingRequests*** table accordingly [5] [6]
    - Update companies' account balances accordingly in the ***companyAccount*** table [5]
    - Provide alerts for overdue requests from the ***requestReceived*** table [7]

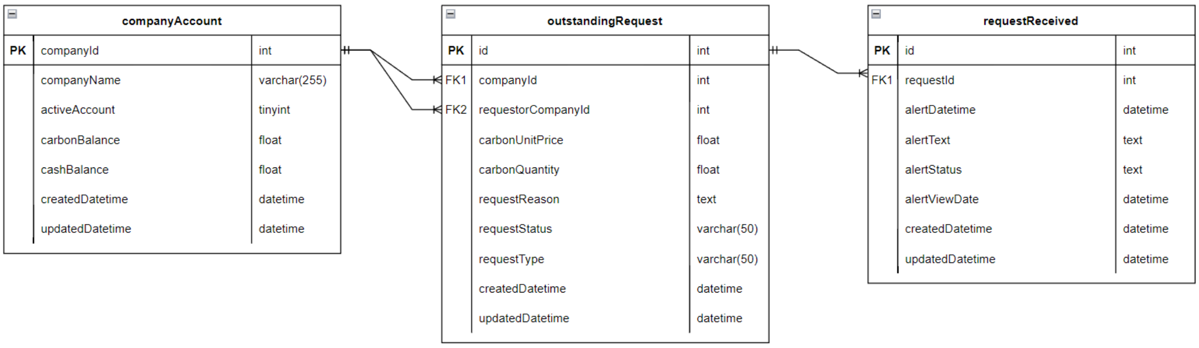
Data Provided

You will be provided:

1. Data in SQL format
    a. Data provided are meant to be for testing purposes and are non-exhaustive.
    b. Tables provided are meant to be a guide. You may add additional columns / attributes or tables to suit your application needs.
2. Database Example
    a. One set of Sample Data provided in tabular plain text format, within this document.

Entity Relationship Diagram (ERD)

The following diagram is provided as a reference for the data tables provided relevant to the requirements given. We have provided sample data for the ***companyAccount, outstandingRequest*** *and* ***requestReceived*** entities.



companyAccount

PK	companyId	int
	companyName	varchar(255)
	activeAccount	tinyint
	carbonBalance	float
	cashBalance	float
	createdDatetime	datetime
	updatedDatetime	datetime

outstandingRequest

PK	id	int
FK1	companyId	int
FK2	requestorCompanyId	int
	carbonUnitPrice	float
	carbonQuantity	float
	requestReason	text
	requestStatus	varchar(50)
	requestType	varchar(50)
	createdDatetime	datetime
	updatedDatetime	datetime

requestReceived

PK	id	int
FK1	requestId	int
	alertDatetime	datetime
	alertText	text
	alertStatus	text
	alertViewDate	datetime
	createdDatetime	datetime
	updatedDatetime	datetime

Database Example

Company Account

(PK) companyId	2025
companyName	TechTrek 2025 Pte Ltd
activeAccount	1
carbonBalance	200
cashBalance	150000
createdDatetime	11/01/2025 09:00:00
updatedDatetime	11/01/2025 09:00:00

Outstanding Request

Note that in this example, company 2025 (requestor) wishes to Sell 3.5 units of Carbon credit at the price of SGD$500.25 to company 2000. It is pending company 2000 to accept this transaction request.

(PK) id	12
(FK) companyId	2000
(FK) requestorCompanyId	2025
carbonUnitPrice	500.25
carbonQuantity	3.5
requestReason	Projected excess carbon credits for 2025
requestStatus	Pending Approval
requestType	Sell
createdDatetime	11/01/2025 09:01:00
updatedDatetime	11/01/2025 09:01:02

Request Received

Note that in this example, a request was received but the alert datetime is not met yet resulting in alertViewDate to be Null.

(PK) id	11
(FK) requestId	12
alertDatetime	18/01/2025 00:00:00
alertText	Overdue request 12: You have yet to approve TechTrek 2025 Pte Ltd's request to sell 3.5 units of carbon at $500.25.
alertStatus	Scheduled
alertViewDate	Null
createdDatetime	11/01/2025 09:01:00
updatedDatetime	11/01/2025 09:01:02

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) ICT Infrastructure
4) Business Strategy Support
5) Data Engineering

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

Do remember that these Extension Modules are meant for you to showcase your domain knowledge. You will be presenting your thoughts and decisions to the assessors, so remember to take into account their suggestions/clarifications.

Pillars	Modules	Requirements
Application Development and Support	**DEV 1**	Application Support
DevOps and Site-Reliability Engineering	**SRE 1**	Availability, Mean Downtime & SRE Principles
ICT Infrastructure	**ICT 1**	Dockerization
	ICT 2	Cloud Technology
Business Strategy Support	**BSS 1**	Project Planning
	BSS 2	Acceptance Test
Data Engineering	**DE 1**	Data Treatment and cleaning

Application Development and Support Modules

[DEV 1]

As an Application Developer, there is a need for efficient development processes and post-production support. For this challenge, you are an Application Support Lead. You are tasked to highlight methods to develop, maintain and improve the applications described by the Main Challenge Statement.

Requirements:

Some things to consider includes (and are not limited to):

1. Security Vulnerabilities and Concerns
2. Performance Optimization
3. Integration considerations and timeline
4. Feature Enhancements / How to push out improvements
5. Version Control

Remember to justify your thought processes clearly and to the best of your ability.

DevOps and Site-Reliability Engineering Modules

[SRE 1]

Availability and Downtime are common challenges for development of any software. Additionally, it is important to design and promote a service management strategy that works for your developed product to build a functional application. For this challenge, you are a Site-Reliability Engineer. You are tasked to consider the issues of Accessibility, Mean Downtime for the application, and apply Site-Reliability Engineering (SRE) principles using measurable objectives.

Requirements:

Some things to consider includes (and are not limited to):

1. How to measure Accessibility
2. How to measure Mean Downtime
3. Factors to decrease Mean Downtime
4. Identify Service Level Objectives and Indicators
5. Develop risk acceptance and mitigation plan
6. How to implement/manage Redundancy and Failover

Remember to justify your thought processes clearly and to the best of your ability.

ICT Infrastructure Modules

[ICT 1]

With increased interest in cloud technology, Dockerization has developed to promote consistency and modularity between application development and cloud deployment environments. For this challenge, you are a Cloud Engineer Specialist. You are tasked to look into the application of Docker technology for an application described by the Main Challenge Statement.

Requirements:

Some things to consider includes (and are not limited to):

1. Containerized architecture
2. Requirements for the application to be Dockerized
3. Pros and Cons of Dockerization of an application
4. Steps should an update of the application be required

Remember to justify your thought processes clearly and to the best of your ability.

[ICT 2]

Cloud services have seen an increased interest over the past few years, with increased developers using cloud technology to deploy their applications. For this challenge, you are a Cloud Engineer Specialist. You will be looking into deploying and managing your application through cloud platforms like 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

1. Pros and Cons of the chosen cloud technology
2. Which option is the most cost effective
3. Which option is the best fit for the applications
4. Scalability
5. Disaster recovery

Remember to justify your thought processes clearly and to the best of your ability.

Business Strategy Support

[BSS 1]

As a Business Strategy Support, there is a need for proper planning for successful application delivery. For this challenge, you are the Project Planning Lead of this project. Not only do you need lead a project effectively, but also demonstrate your ability to ensure its resource optimization, cost-efficiency, and technical robustness.

Requirements:

Some things to consider includes (and are not limited to):

1. Scope definition
2. Manpower Allocation
3. Timeline and Project Risk Assessment

Remember to justify your thought processes clearly and to the best of your ability.

[BSS 2]

Thorough testing plans are vital to ensure that the application's quality and readiness. Additionally, it is important to validate that the application is aligned with user requirements. For this module, you must establish the testing lifecycle for your application.

Requirements:

Some things to consider includes (and are not limited to):

1. Quality Assurance Integration
2. Test Objectives, Scope and Acceptance Criteria
3. Unit / Integration Tests
4. Test coverage and metrics
5. Defect Management

Remember to justify your thought processes clearly and to the best of your ability.

Data Engineering

[DE 1]

As the world becomes more digitized, data is generated rapidly across multiple platforms. In order to make use of these data productively, proper data treatment is necessary before storing and generating insights. In this module, you will be tasked to perform data treatment and build simple data pipelines to integrate into your project. You may refer to the provided link for data API or to download data. It is recommended that you perform this task in Python or on an Excel spreadsheet.

Requirements:

You may make use of the data from https://www.climatewatchdata.org/data-explorer/historical-emissions via the API or by downloading the data from the page. Do filter and select *all sectors* before proceeding.

Some things to consider includes (and are not limited to):

1. Data Profiling
2. Data Cleaning / Normalization
3. ETL processes
4. Error handling
5. Data Validation
6. Scalability

Remember to justify your thought processes clearly and to the best of your ability.