

TechTrek 2025

Challenge Statement (Addendum)

11th January 2025

Breaking World News

The TechTrek Federal Government announced an immediate imposition of a 3000% increment to the current worldwide Carbon Tax, affecting all companies. This policy mandates heavy penalties for excessive carbon emissions, pushing companies to reduce emissions or *offset them using carbon credits*.

All corporations are now scrambling to comply with the new regulations, with many companies aggressively hoarding carbon credits and creating market scarcity. Prices now surge to unprecedented levels, incentivizing credit holders to sell at extremely high margins.

The unprecedented surge in carbon credit prices has exposed the limitations of traditional trading platforms, which struggle to accommodate the rapid pace and complexity of transactions. To address this, all platforms are transitioning into **bidding-based systems**, allowing buyers to compete dynamically for credits in real-time.

Challenge Statement

Your task is to **pivot** your trading platform designs towards a carbon credit **bidding** platform that allows users, such as companies, to **bid** for carbon credits. The application should offer essential features to allow companies to list their carbon credits in a global marketplace where buyers can place bids. This must be a **web application**.

It is not required to develop a fully functional system. Pseudocode or a detailed explanation of your design process incorporated into your final presentation are also acceptable.

All the requirements included below are meant to be a rough guideline to guide your team on how to tackle this challenge statement. Do communicate actively with your assessors for any other additional areas to consider.

Basic Application Requirements (Integrate):

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

Basic Application Features:

1. **Bid Placement:** Buyers should be able to place bids on carbon credits they wish to purchase. They can set their desired price and quantity, and the system will track their bids in real time.
2. **Bid Matching:** When a seller offers credits that match the buyer's bid, the system should automatically notify both parties. Once the bid is accepted, the transaction proceeds with the agreed-upon price.
3. **Auction Style or Reverse Auction:** The platform should support either a traditional auction (where sellers set the initial price and buyers bid up) or a reverse auction (where buyers set the price and sellers respond). Candidates are free to select the auction style of their choice.
4. **Bidding Deadline and Notifications:** Set time limits for bidding. with countdown timers and notifications to inform users when a bid is about to expire or if they've been outbid.
5. **Transparency and Fairness:** Ensure that all bids are transparent and that users can view the bid history and pricing trends to make informed decisions.
6. **Transaction Finalization:** Once a bid is successful, the platform should ensure a smooth transaction process, including payment handling, credit transfer, and confirmation to both buyer and seller.

*These requirements are for your reference should you choose to pseudocode/develop the application.

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
Bidding Page	[2]	Display company's outstanding balances for: <ul style="list-style-type: none"> Carbon Credits Cash Balances 	Return user's balance details from companyAccount table: <i>companyName, carbonBalance and cashBalance.</i>
	[3]	Display all outstanding bids listed by your company with the following details: <ul style="list-style-type: none"> Bidding Window (Deadline) Carbon Bid Price (SGD/Tonnes) Carbon Bid Quantity Bid Type (Buy/Sell) Auction Style Highest Bid for bid window 	Return a list of outstanding bid listings from the outstandingBids and company table: <i>companyName, carbonBidUnitPrice, carbonBidQuantity, auctionStyle, bidDeadline and bidType.</i>
	[4]	Insert/Edit/Delete bids listed by your company with the following details : <ul style="list-style-type: none"> Bidding Window (Deadline) Carbon Bid Price (SGD/Tonnes) Carbon Bid Quantity Bid Type (Buy/Sell) Auction Style 	Able to create new bid listings, edit the listing and delete the listing from outstandingBids table. Creating requests should also update the bidReceived table.
Bidding Window Page	[5]	Display all outstanding bid listings made from all companies, sorted by bid prices with the following details: <ul style="list-style-type: none"> Bidding Window (Deadline) Bidding Company Name Carbon Price (SGD/Tonnes) Carbon Quantity Bid Status Bid Type (Buy/Sell) Auction Style 	Return a list of outstanding bid listings from the outstandingBids and companyAccount tables: <i>CompanyName, carbonBidUnitPrice, carbonBidQuantity, auctionStyle, bidStatus, bidDeadline and bidType.</i>
	[6]	Bid Matching; Notifies users when there are sellers offering credits that matches buyer's bids. Button for buyers to propose a price during bidding window for reverse auction style.	Update the database for outstandingBids table to indicate bid status once there are bid matches depending on auction style. Process bids and make changes to companyAccount table.
	[7]	Provide Countdown Timers and Notifications for current bidding window (set as 1 calendar day).	Returns bid notifications and corresponding bidding window from outstandingBids and bidReceived tables.

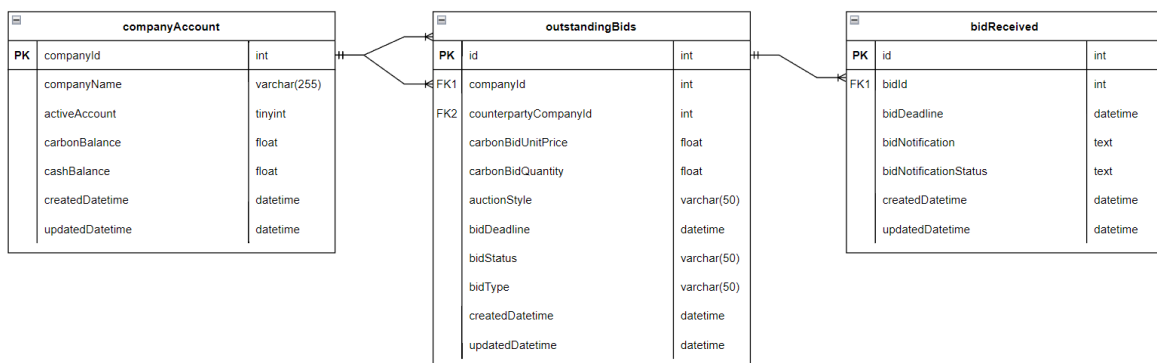
Data Provided

You will be provided:

1. Limited 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 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**, **outstandingBids** and **bidReceived** entities.



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 Bids

Note that in this example, company 2025 (successful bid) purchases Sell 3.5 units of Carbon credit at the price of SGD\$500.25 from company 2000. This is also under Traditional Auction Style, the bid was system-matched and the transaction will be processed soon.

(PK) id	12
(FK) companyId	2000
(FK) counterpartyCompanyId	2025
carbonBidUnitPrice	500.25
carbonBidQuantity	3.5
auctionStyle	Traditional
bidDeadline	11/01/2025 23:59:59
bidStatus	Matched
bidType	Sell
createdDatetime	11/01/2025 09:01:00
updatedDatetime	11/01/2025 09:01:02

Bid 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	1
(FK) bidId	28
bidDeadline	11/01/2025 23:59:59
bidNotification	Bid Window Closing. There are no bid matches for bid 28.
bidNotificationStatus	Scheduled
createdDatetime	11/01/2025 11:30:00
updatedDatetime	11/01/2025 23:59:00