

E-AUCTION APPLICATION

CONTENTS

1	Problem Statement	2
2	Proposed E-Auction Application Wireframe	2
3	Business Requirements:	3
4	Proposed Rest Endpoints to be exposed	6
4.1	Rest APIs:	6
5	EXPECTED DELIVERABLES	6
5.1	ENGINEERING CONCEPTS (SECURITY & IDENTITY):	6
5.2	PRODUCTS & FRAMEWORK (DATABASE & STORAGE):	6
6	PLATFORM	6

1 PROBLEM STATEMENT

E-Auction Application is microservice based Cloud Native Application. The Main purpose of **E-Auction** is to provide the ability to initiate sale, purchase and bid for a product by the customer. The core modules of E-Auction app are:

- ✓ Put a new product on auction with starting bid price.
- ✓ Bid for the product.
- ✓ Update the bid price.
- ✓ Enlisting of bids placed on any product.

Use Flutter to design Mobile App.

The scope includes developing the application using tool chain mentioned below.

2 PROPOSED E-AUCTION APPLICATION WIREFRAME

Below is the wireframe for reference.

LOGO	Fetch Details		
Product	<input type="text"/>	GET	
Product Name	<input type="text"/>		
Short Description	<input type="text"/>		
Detailed Description	<input type="text"/>		
Category	<input type="text"/>		
Starting Price	<input type="text"/>		
Bid End Date	<input type="text"/>		
Bids			
Bid Amount	Name	Email	Mobile
1000/-	First	first@mail.com	1234567890
1500/-	Second	second@mail.com	2345678901
800/-	Third	third@mail.com	3456789012
1200/-	Fourth	fourth@mail.com	4567890123

3 BUSINESS REQUIREMENTS:

Below are the user stories for the given problem statement

User Story #	User Story Name	User Story	Development
US_01	Seller Functionalities	<p>As a seller I can add a new product for auction</p> <p><i>While adding a product, following information is required.</i></p> <ul style="list-style-type: none">➤ Product Name➤ Short Description➤ Detailed Description➤ Category➤ Starting Price➤ Bid end date <p>It also requires seller information as follows:</p> <p>Personal Information</p> <ul style="list-style-type: none">➤ First Name➤ Last Name➤ Address➤ City➤ State➤ Pin➤ Phone➤ Email <p>Constraints:</p> <ol style="list-style-type: none">1. Product Category should be the one from the existing. (Predefined Hard Coded)<ol style="list-style-type: none">a. Paintingb. Sculptorc. Ornament2. Starting price should be number3. Bid end date should be future date, else throw a custom exception in such case <p>Validations:</p> <ol style="list-style-type: none">1. Product Name is not null, min 5 and max 30 characters.2. firstName is not null, min 5 and max 30 characters.	Only API to be developed

		<ol style="list-style-type: none"> 3. lastName is not null, min 3 and max 25 characters. 4. email is not null, and it should be valid email pattern, containing a single @. 5. mobile is not null, min 10 and max 10 character and all must be numeric. 	
US_02	Seller Functionalities	<p>As a seller I can delete a product added for auction</p> <p>Constraints:</p> <ol style="list-style-type: none"> 1. If product is tried to be deleted after the bid end date, it must throw a custom exception. 2. When product is tried to be deleted and if at least one bid is placed on that product, throw a custom exception. 	Only API to be developed
US_03	Buyer Functionalities	<p>As a buyer I am able to bid for a product</p> <p><i>While bidding for a product, following information is required from buyer.</i></p> <ul style="list-style-type: none"> ➤ First Name ➤ Last Name ➤ Address ➤ City ➤ State ➤ Pin ➤ Phone ➤ Email ➤ Product Id ➤ Bid Amount <p>Precondition:</p> <ol style="list-style-type: none"> 1. ProductId must be of an existing product available for auction (Can get from Database) <p>Constraints:</p> <ol style="list-style-type: none"> 1. If bid is placed after the bid end date, it must throw a custom exception. 2. More than one bid on a product by same user (based on email ID) is 	Only API to be developed

		<p>not allowed, must throw a custom exception in such case.</p> <p>Validations:</p> <ol style="list-style-type: none"> 1. firstName is not null, min 5 and max 30 characters. 2. lastName is not null, min 3 and max 25 characters. 3. email is not null, and it should be valid email pattern, containing a single @. 4. mobile is not null, min 10 and max 10 character and all must be numeric. 	
US_04	Seller functionality	<p>As a seller I can list all bids received on product put for auction</p> <p>Considerations:</p> <ol style="list-style-type: none"> 1. While fetching the above details the API must return the product details (Short Description, Detailed Description, Category, starting price, Bid End Date) along with all bids placed on it. 2. Bid details must be in descending order of bid amount. 	API and Frontend to be developed
US_05	Buyer functionality	<p>As a buyer I am able to update the Bid Amount of my bidding</p> <p>Constraints:</p> <ol style="list-style-type: none"> 1. If bid Amount is updated after the bid end date, it must throw a custom exception 	Only API to be developed

4 PROPOSED REST ENDPOINTS TO BE EXPOSED

4.1 REST APIs:

URL Exposed	Purpose	
/e-auction/api/v1/seller/add-product	Adds a new product	
/ e-auction/api/v1/seller/show-bids/{productId}	Fetches details of all bids on product with product details	
/ e-auction/api/v1/seller/delete/{productId}	Deletes the product	
URL Exposed	Purpose	
/e-auction/api/v1/buyer/place-bid	Places a bid for a product	
/ e-auction/api/v1/buyer/update-bid/{productId}/{buyerEmailId}/{newBidAmount}	Updates the bid amount	

5 EXPECTED DELIVERABLES

5.1 ENGINEERING CONCEPTS (SECURITY & IDENTITY):

1. As an Application Developer:
 - a. Restrict the access over all write operation (secured operations) by adding authentication
 - b. Secure all Rest EndPoints by configuring SSL Certificate for Cloud

5.2 PRODUCTS & FRAMEWORK (DATABASE & STORAGE):

1. As an application developer:
 - a. Ensure Application is implementing ORM with .Net Core
 - b. Use SQL SERVER for maintaining

6 PLATFORM

Use above Business Requirements to implement the below.

1. Use Azure to deploy application on cloud.
2. Use Azure SQL Server as a database for the Application.
3. Use Azure Functions and DB to build a backend process for handling requests for E-Stock Market App.
4. Use Azure Service to send email to company after stock is listed.

Note: Minimum 2APIs (UI+Backend) to be hosted in cloud