**Car-Auction On Blockchain Using HyperledgerFabric**

In this project one can list assets for sale (setting a reserve price), and watch as assets that have met their reserve price are automatically transferred to the highest bidder at the end of the auction.

* **Project Link**
* <https://github.com/manu461/ProjectsDemo/tree/master/BlockChainProjects/HyperLedgerComposerFabricProject/carauction-network>
* <https://github.com/manu461/ProjectsDemo/tree/master/BlockChainProjects/HyperLedgerComposerFabricProject/carauction-network/carauction-angular-app>
* **Project Description**

I created this project while completing **IBM Blockchain-Foundation Developer** course.

Car-Auction’s(this project’s) Business-network is developed using **Hyperledger Composer**, whereas the network been deployed locally using **Hyperledger Fabric**.

The complete package**(.bna)** containing **Business-Model**(.cto file), **script files**(.js file), **ACLs**(.acl file) and **Metadata**(.md file) is converted to an **Angular Application** using **YO code generator**.

In this project one can list assets for sale (setting a reserve price), and watch as assets that have met their reserve price are automatically transferred to the highest bidder at the end of the auction.

This business network defines:

1. Participants
2. Member
3. Auctioneer
4. Assets
5. Vehicle
6. VehicleListing
7. Transactions:
8. Offer
9. CloseBidding

The **makeOffer** function is called when an **Offer** transaction is submitted. The logic simply checks that the listing for the offer is still for sale, and then adds the offer to the listing, and then updates the offers in the **VehicleListing** asset registry.

The **closeBidding** function is called when a **CloseBidding** transaction is submitted for processing. The logic checks that the listing is still for sale, sorts the offers by bid price, and then if the reserve has been met, transfers the ownership of the vehicle associated with the listing to the highest bidder. Money is transferred from the buyer's account to the seller's account, and then all the modified assets are updated in their respective registries.

\*\*NOTE: This is not my idea to develop such project, it is already provided as a template for business-network by IBM at their composer-playground website. I have studied the code and logics thoroughly and implemented it during completing IBM Blockchain-Foundation Developer course.

* **Tools Used**

1. Data-Modelling is done with **composer modelling language**. It is domain specific language for describing the nature of the business network. It is developed by IBM.
2. Business-logic is written in **JavaScript**.
3. **Composer-Playground** is used to do ideation and very quickly prototype composer solution.
4. Two main **npm** modules
5. Composer-client
6. Composer-Admin

which can be embedded in application for programmatic access.

1. **VSCode** is used as code-editor.
2. **CLI tools**
3. Complete package of Business network Archive (.bna) is converted into an Angular-Application using **YO code generator**.

* **Project Walkthrough**

1. **Starting the Application**