|  |  |
| --- | --- |
|  | **HASAN KALYONCU UNIVERSITY**  **Computer Engineering Department** **COME 499 Project Proposal Form** |

**Part I. Project Proposer**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name Lastname** | **Mustafa BIÇAKCI** | **E-mail** | **mustafa.bicakci@hku.edu.tr** |

**Part II. Project Information**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Starting Term** | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 2 | 0 | 2 | 0 | / | 2 | 0 | 2 | 1 | |
| **Title of the Project** | Online auction application |
| **Project Description** | |
| In an online auction, buyers and sellers engage in transactional business, wherein buyers purchase items through price bidding. Here, the bids have a starting price and an ending time. Potential buyers who place the highest bidding price for an item are declared the winners and owners of particular items.  In this project, you will create a secure online auction dashboard using the fraud detection method with binary classification. If a user wants to buy a product through an online auction, they must provide their identification details like Phone number email address, TC number etc. The system will then screen the users, authenticate, and authorize them. Only authorized users can bid in the auction. The system will be designed to predict fraudulent users in the early stages, thereby eliminating the risk of online fraud and scams. | |
| **Project Justification** | |
| **Novelty** | |
| **New aspects** | Creating a high level secure dashboard application using the latest programming techniques and design patterns. |
| **Complexity** | |
| **Challenging problem and issues** | Using two factor authentication to reduce scam users and using an online payment api (stripe) to receive online payment in real time. |
| **Related computer science fields and subfields** | Software design patterns, programming framework, database design, user experience and user interface development, fraud systems, online payment methods. |
| **Tools** | Tech Stack, Django Framework, Python, Two-Factor Authentication, Stripe payment APIs, Postgres,  Front end Stack: JavaScript (React or Angular)  Back-end: Python (Django / Flask)  Databases: PostgreSQL or MongoDB |
| **Risk involved** | |
| **Potential problems and alternative solutions** |  |
| **Minimum work required** | 14 weeks with 3 students. |