SCOPE OF PROJECT: Office space and facilities micro-renting broker assistant

PORTAL WEB

1. Landing:
    - Create account (direct)
    - Create account with FB (open id)
    - Create account with Google (open id)
    - Login
    - Privacy & confidentiality link -> text page in new tab
2. Create account:
    - Fields:
        i. Username, Name, email, company, telephone, CUI, company address, IBAN and bank
        ii. Account type
            - Supplier: setup **Profile (**or "default keywords filter"): by **searching** in static pre-defined list **(editable in "view profile")**
                - Exemplu: *#curatenie #renovare #secretara #instalatiisanitare*
            - Customer
            - Public (views requests only)
    - Password: **(editable in view profile)**
        i. Encrypted save
        ii. Password strength meter (basic free)
3. View profile:
    - View all fields from account creation - IBAN is partially obfuscated (example: RO40BAxxxxxxx0134)
    - Edit password
    - Edit profile keywords (only for supplier)
    - Delete account
4. Supplier Dashboard (2 views)
    - (Main) List ONLY projects with default filter ("default keywords filter")
        i. Search by keywords in filtered projects (you can reduce the number of keywords: say only *#renovare* or *#renovare#instalatiisanitare*)
        ii. Apply to project - > go to "my project": when applying the suppliers selects the specific tags (allowed from the profile) that he wants to apply to (see below example) and also completes a **small text note** to attach to the application
        iii. Example:
            - `PROIECT XXX: #bucuresti, #inchiriere, #copycenter, [#curatenie]`

- `PROIECT XYZ: #iasi, #inchiriere, #1000mp,` **`[#secretara], [#curatenie]`**
- `PROIECT XZZ: #cluj, #inchiriere, #secretara, #curatenie,` **`[#renovare]`** **-** *in this case the Supplier only selected #renovare out of the allowed profile keywords*

○ My projects:

i. Archived projects (closed history projects list)

ii. Active projects (un-signed project list with chosen components / all components)

- Bid but not closed (waiting for Customer to deliberate and notify)
- Closed with a win but not archived (customer accepted, must upload file/clarifications, bank letter, etc)

5. Customer Dashboard (3 views):

○ "New project" -> conversation -> "project profiling keywords" review -> submit -> **new project visible to all suppliers that have at least 1 keyword**

i. The user will communicate with the chatbot (which runs on an embedded system - Jetson) through a REST API - after each user's utterance:

- The request will contain the whole conversation until that point - both user's utterances and chatbot's utterances;
- The response will contain the next chatbot's utterance, as well as the list with all tags (the conversation summarization) until that point

ii. If not submitted, then the user will be returned to the initial view ("New project")

○ Project status

i. % completed of all keywords in "project profiling keywords"

ii. Wait for 100% tags to be assumed by suppliers

iii. Assign winner on multi-supplier keyword -> view competitive suppliers on the same tag/tags + see their bid notes -> select winner

○ Completed projects:

i. View all completed projects

ii. Generate contract -> html .doc file (universal HTML with tags/fields where we can replace with field data)

- Parts of the contract
- The object of the contract -> auto-completed based on the tags, or eventually, based on a dictionary that describes each tag

iii. Upload additional file(s)

iv. Notify suppliers -> automated email to supplier address including generated contract and uploaded file(s)

6. Admin dashboard
    - List all projects
        i. Project ID /// tag1 / tag2 / tag3 …. /// state (active or completed) (headline-ul)
            - View conversation (popup) -> button “send to AI retrain” (nu face nimic)
            - View suppliers -> foldable
                - If project is active, show current bidders
                - If project is completed, show winners
    - View all users
        i. Customers
            - List of all customers
            - When choosing a customer, view logins history
        ii. Suppliers
            - List of all suppliers
            - When choosing a supplier, view logins history
    - Reports (external python scripts that have access to the db):
        i. Report 1 (distribution of tags per requests) : external python script 1 call -> HTML file which should be rendered in a new tab (python script Damian)
        ii. Report 2 (...) : external python script 2 call -> HTML file which should be rendered in a new tab (python script Damian)
        iii. Maximum 5 reports
7. Other:
    - (optional) Python language
    - (optional) responsive UI
    - (mandatory) persistent data store (db)
        i. Users information including nr unsuccessful authentications
        ii. Separate table for financial information (IBAN and bank)
        iii. “Project” table
        iv. History of all logins in the platform for each user
            - USER_ID
            - TIME
            - IP
            - LOCATION
    - (mandatory) standard web-server log - IP addresses and geographic locations
    - (mandatory) Security measures:
        i. Auto logout the user after a specific time interval
        ii. Brute force attacks detection and prevention by limiting the number of unsuccessful authentications.

- After reaching the maximum nr of unsuccessful attempts, the platform will not allow the user to authenticate in the following N minutes (N to be defined)

- (mandatory) db accessible from external python scripts (see reporting)
- **Project states:**
  i. **Consumer: Conversation (pre-project)**
  ii. **Consumer: Tags (pre-project)**
  iii. **Consumer: New project submitted**
  iv. **Supplier: apply to tags (click and write short note)**
  v. **Consumer: Wait for all tags to complete**
  vi. **Consumer: Approve each Supplier (including competing suppliers)**
  vii. **Supplier: upload additional information (bank letter) after receiving approval**
  viii. **Consumer: send contract**
- **"Project" table** fields**:**
  i. **ID**
  ii. **Consumer**
  iii. **Name**
  iv. **Project state (see above)**
  v. **ALLAN Conversation (private to Consumer & admin)**
  vi. **Tags**
  vii. **List Supplier-tag + note**
  viii. **Supplier files**
  ix. **Contract**

GITHUB

1. Code
2. Complete installation and running documentation:
   - How to install the dependencies
   - Starting the portal engine both in local environment and in production environment (ex: VM with NGINX)