April 2022

Refactory

Catalyst 10

Final Technical Assessment

REFACTORY CATALYST10 FINAL TECHNICAL ASSESSMENT

Introduction:

The world is was recently faced with a pandemic of Coronavirus Disease (COVID-19), a virus that claimed lives of very many people in Uganda and the world at large. As the world grapples with the post-covid economic revival, People's Financial statuses of course must be the first level of concern, with focus on redeeming and rejuvenating People's income stability for their financial sustainability.

Uganda is at the fore front of pushing for this agenda and one of the very active actors in this case is Uwezo Finance services. Its an Entity whose sole objective if to provide affordable finance solutions to Ugandans, services like Loans, Asset financing, and Financial Literacy.

Challenge:

In attempt to simplify their service delivery, The loans department of Uwezo Finance Services Proposed an electronic Information system to smoothen their work of trying to ease the delivery of loans in Uganda. The proposal, was approved, the system was documented, planned, and designed as well, and now all efforts are directed towards implementation of the system. You are one of the selected qualified and technically competent full stack Software developers and You have been assigned the "Loan Registration form" module to be used for registration of whoever apply for a loan. Prove your competence by honoring and fully addressing the challenge presented to you.

Proposed Module development time:

Total Duration: 4Hours

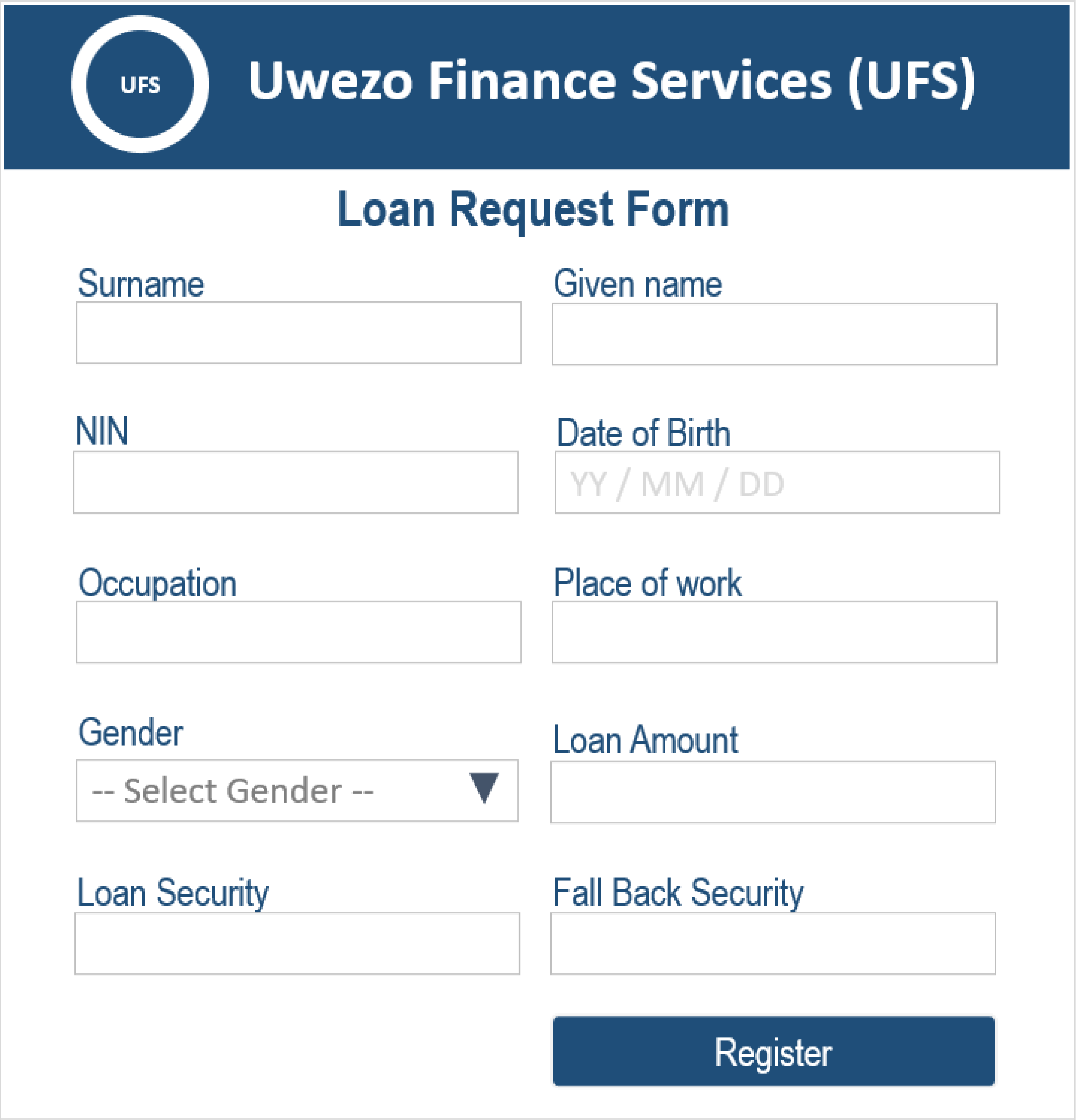
Requirements [What to be Implemented]:

- System should render the registration form upon request.
- User can enter details of Client into the registration form.
- User can submit the registration form (with Client details)
- Upon registration-form submission, the system should validate all fields.
- Upon registration-form validation, if any field is invalid, the system should Inform the user about the invalidity and terminate the submission process.
- After registration form validation, the system should submit a form only if all fields are valid.

- After registration-form submission, the system's Resource server should receive the form data and store it in the database.
- After a successful registration form submission, Validation and data storage, the system should redirect to the same form or reset the form and notify the user with a success message.
- The success message should disappear upon close or upon focus of any of the form elements.

Proposed User Interfaces according to the form designs:

According to the designs of the proposed system, after requesting for the Registration form from the server,



UFS

Uwezo Finance Services (UFS)

Loan Request Form

Surname

Given name

NIN

Date of Birth

YY / MM / DD

Occupation

Place of work

Gender

-- Select Gender -- ▼

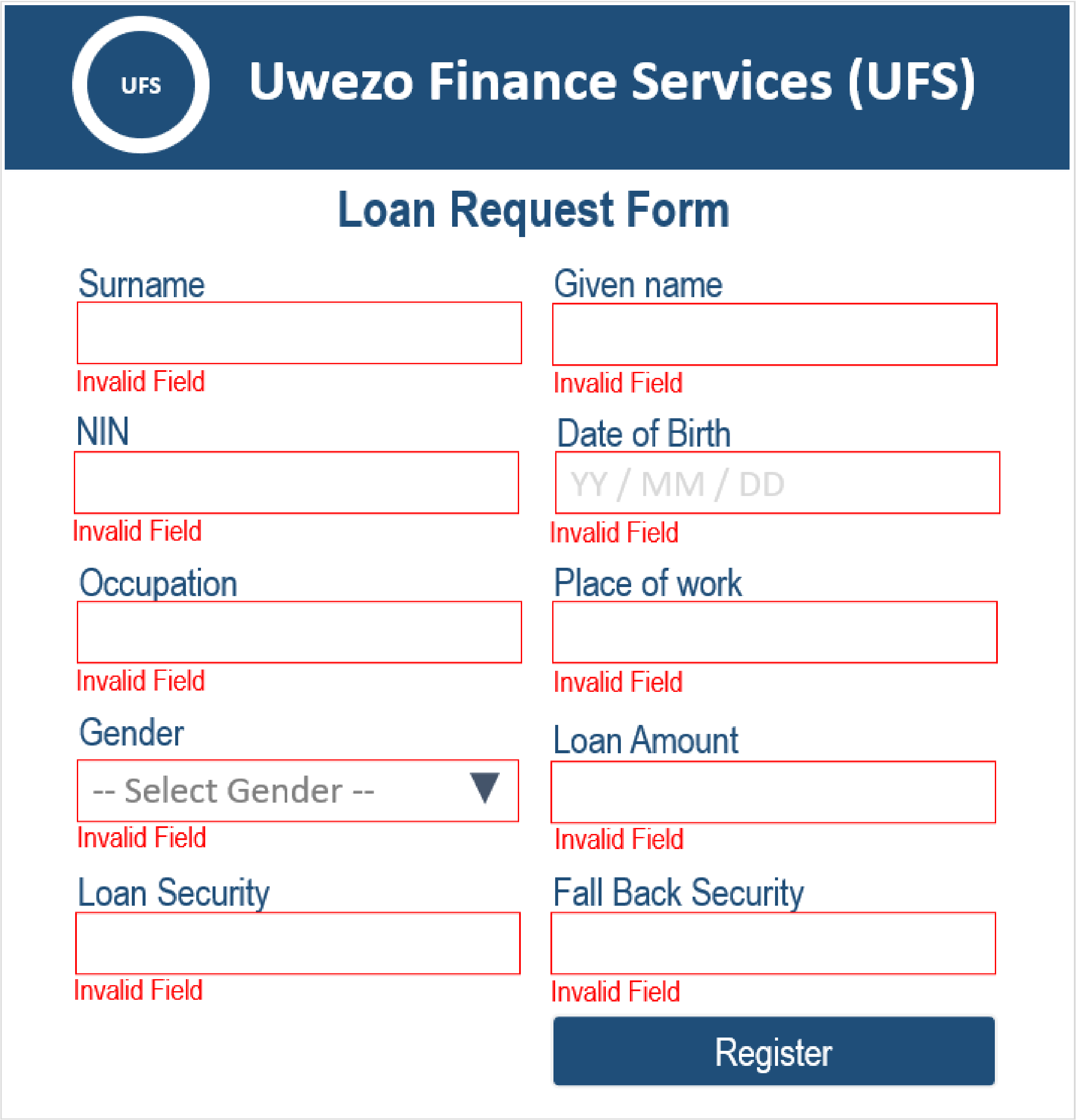
Loan Amount

Loan Security

Fall Back Security

Register

Below is the proposed look for each of the invalid fields after form validation.



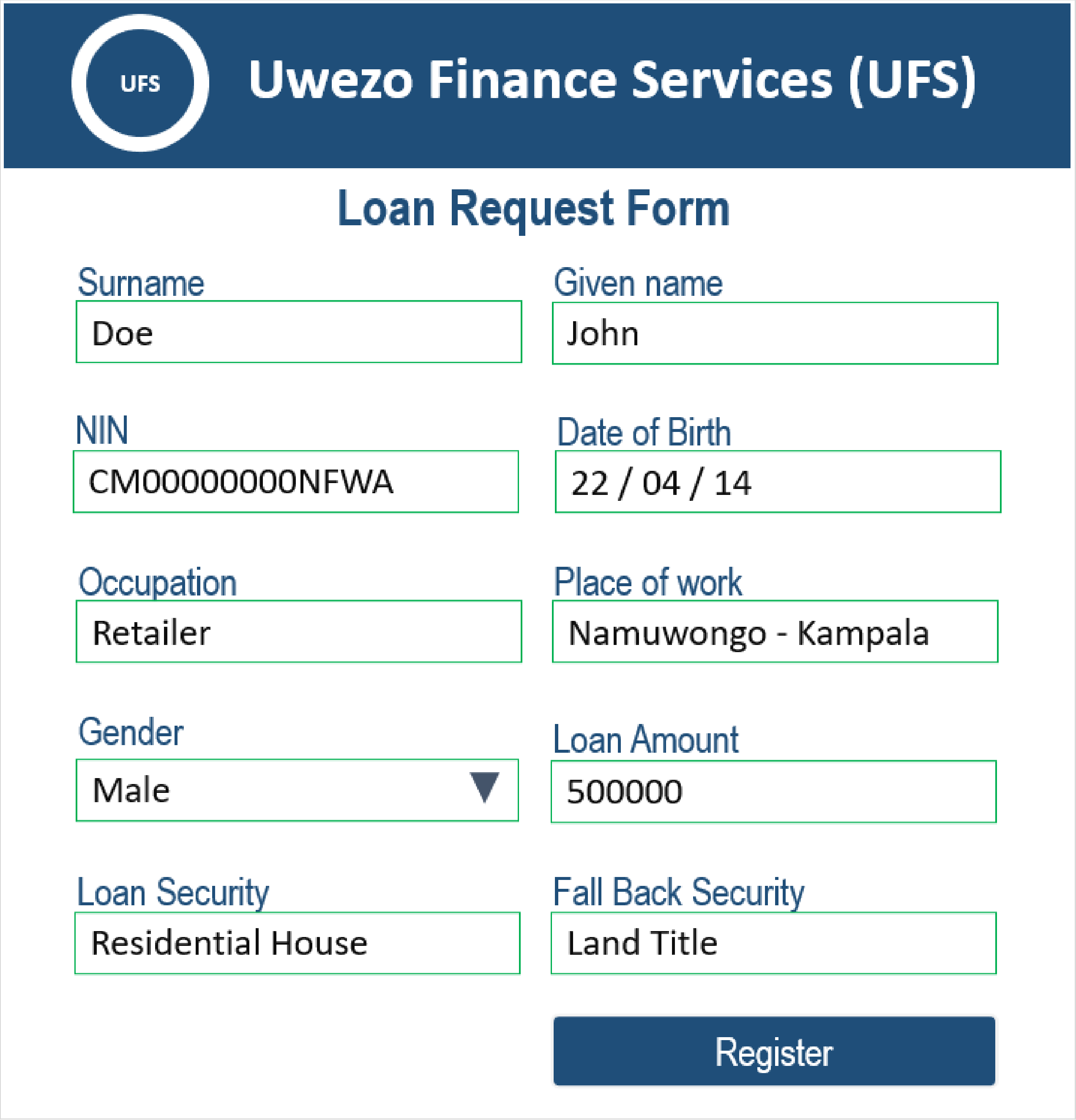
UFS

Uwezo Finance Services (UFS)

Loan Request Form

Surname Invalid Field	Given name Invalid Field
NIN Invalid Field	Date of Birth YY / MM / DD Invalid Field
Occupation Invalid Field	Place of work Invalid Field
Gender -- Select Gender -- ▼ Invalid Field	Loan Amount Invalid Field
Loan Security Invalid Field	Fall Back Security Invalid Field
	Register

Below is the proposed look for the Registration form with valid entries.



Uwezo Finance Services (UFS)

Loan Request Form

Surname: Doe

Given name: John

NIN: CM00000000NFWA

Date of Birth: 22 / 04 / 14

Occupation: Retailer

Place of work: Namuwongo - Kampala

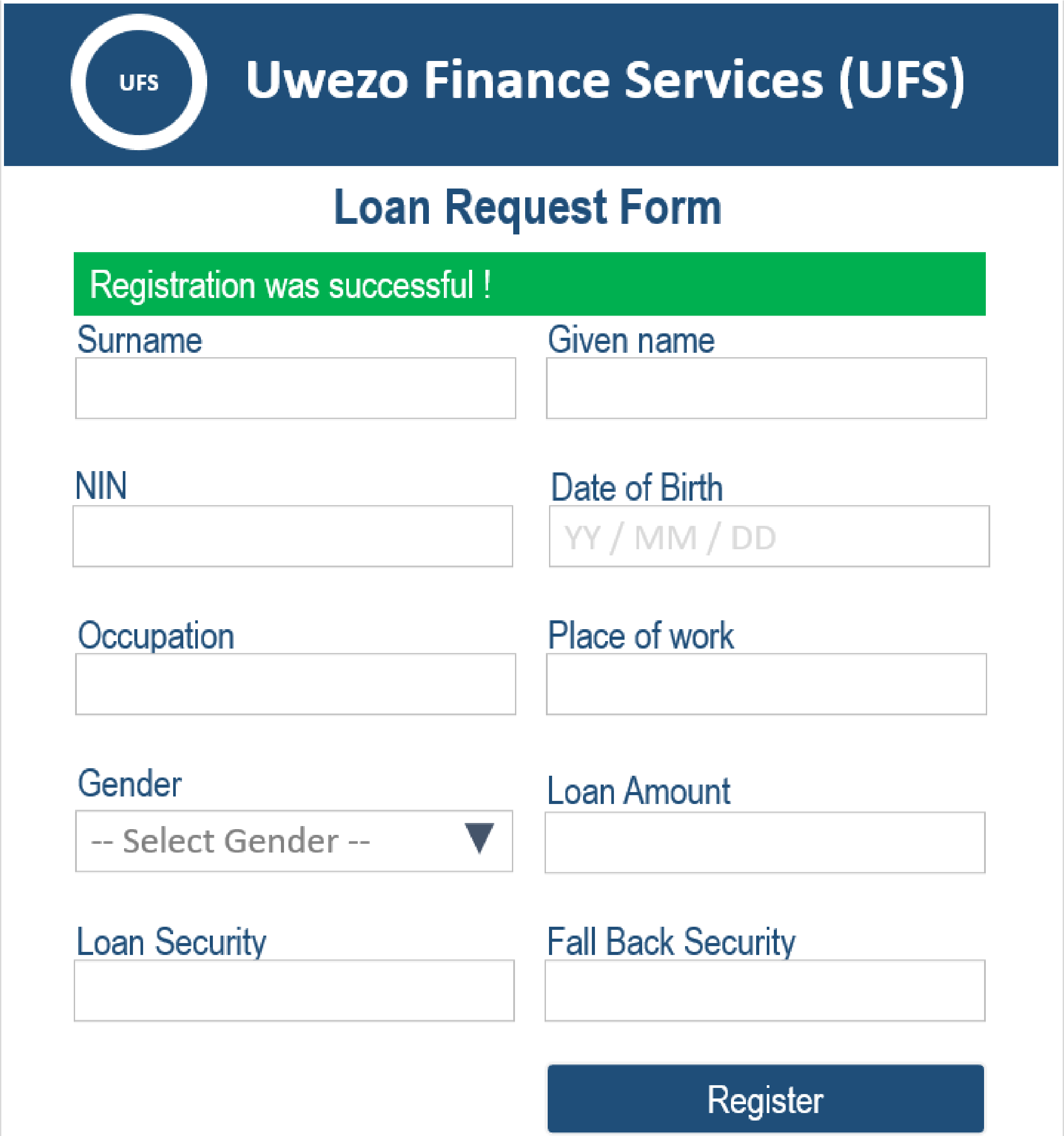
Gender: Male ▼

Loan Amount: 500000

Loan Security: Residential House

Fall Back Security: Land Title

Below is the proposed look for the Registration form immediately after a successful registration of a loan.



UFS

Uwezo Finance Services (UFS)

Loan Request Form

Registration was successful !

Surname

Given name

NIN

Date of Birth

YY / MM / DD

Occupation

Place of work

Gender

-- Select Gender -- ▼

Loan Amount

Loan Security

Fall Back Security

Form Validation Guidelines:

Validation of the Form Fields Should be done basing on the following guidelines:

1. **Surname**

   - Should be a string of not less than 2 characters
   - Should be a string of not more than 255 characters
   - Should not contain a numeric character

2. **Given Name**

   - Should be a string of not less than 2 characters
   - Should be a string of not more than 255 characters
   - Should not contain a numeric character

3. **NIN**

   - Should be a string of not less than 13 characters
   - Should be a string of not more than 14 characters
   - Should start with 2 characters (CM/CF)
   - After the first 2 characters, the next 8 characters should be numeric in nature
   - After the first 10 characters, the rest of the characters should not be numeric in nature.

4. **Date Of Birth**

   - It should be a valid date
   - It should takeon the format (YY/MM/DD),
   - The difference between the date now and the date of birth should not be less than 18 years.

5. **Occupation**

   - Should be a string of not less than 2 characters
   - Should be a string of not more than 255 characters

Form Validation Guidelines:

Validation of the Form Fields Should be done basing on the following guidelines:

6. **Place of work**

    - Should be a string of not less than 2 characters

    - Should be a string of not more than 255 characters

7. **Gender**

    - Should be a selectable field

    - The only available options are [-- Select Gender --, Male and Female]

    - "-- Select Gender --" as an option is only a Place holder

    - The only options valid for submission are [Male, Female]

8. **Loan Amount**

    - Should be a numeric value

    - should not be less than 500000 (Five hundred thousand)

    - should not be greater than 50000000 (50 million)

9. **Loan Security**

    - Should be a string of not less than 5 characters

    - Should be a string of not more than 255 characters

10. **Fallback Security**

    - Should be a string of not less than 5 characters

    - Should be a string of not more than 255 characters

Programming Languages, Libraries, Frameworks and tools to be used:

1. **HTML**
2. **CSS**
   - Bootstrap (optional)

3. **JavaScript**
   - JQuery (optional)

4. **Nodejs**
   - Vue.js (optional)
   - Express
   - Body-Parser
   - Pug (optional)
   - Mongoose
   - Passport.js (optional)
   - bcrypt (optional)

5. **Git**
   - GitHub
   - GitHub Desktop (Optional)

6. **MongoDB**
   - MongoDB Compass

Text Editors and IDEs:

- Visual Studio Code (VScode).

Preparation Guidelines:

Note: Assessment Starts right from here

These are the steps you ought to take to get ready to start building the project

Step 1: Check for whether you have Git installed on the computer you are to use.

Step 2: Login to your GitHub account (Create one if you don't have).

Step 3: Follow this link https://github.com/tech-refactory/Refactory-Catalyst10-Final-Technical-Assessment and fork the repository named **Refactory-Catalyst10-Final-Technical-Assessment**.

Step 4: Now clone a fork (copy) of the above-mentioned repository to any desired location (directory) on your computer (like on the desktop).

Step 5: Open the **Cloned repository** in a text editor and a terminal or Console input and output window (Most preferably VScode with Git Bash as the integrated terminal) Otherwise if you are to use a text editor with no integrated Console Terminal, Open the Cloned repository in any terminal availed by your Operating system, i.e. **CMD**, **POWERSHELL LINUX TERMINAL, etc**.

> **Note:** Every time you are to stage changes, commit changes or push changes to a remote repository, ensure that the path is pointing to this cloned repository.
>
> **Note:** Endeavor not to Initialize any repository within this repository.
>
> **Note:** Ensure to have the gitignore file (if not create one) right at the start of this repository. Use it to ignore all the un-necessary files and Directories with in this repository so you don't make them part of your commits and pushes.

Step 6: Open that cloned copy on your computer and then within it create an empty folder and name it with you First and Last name (or Sur and Given name).

> **Note:** You may receive Instructions from your Invigilator or Instructor in a refined way from what is in this step (Step 6)

Step 7: Within that folder (the one with your name) is where you should have your project done. Now start and work on your project as you add, commit, and push to update your remote repository.

> **Note:** The repository you push changes to should be (a fork) the one you attained after forking the central repository (as in step3). Don't Push to (or try to update) the central repository during development.
>
> **Note:** You are expected to make only one pull request and that is at the end of your assessment. All pull requests will be merged once at the end of the assessment by the Administrator of the central repository.

Success wishes from

The Refactory Technical Department

On behalf of Refactory

refactory