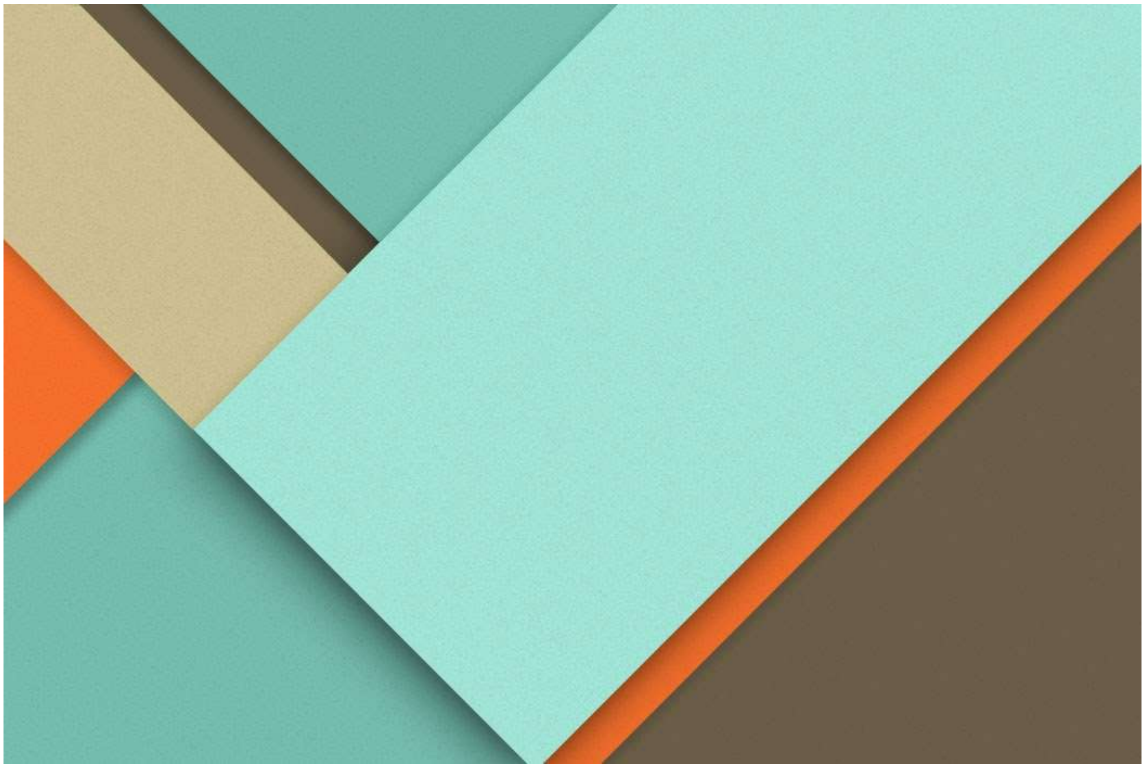


# IFRS 16 – NLS

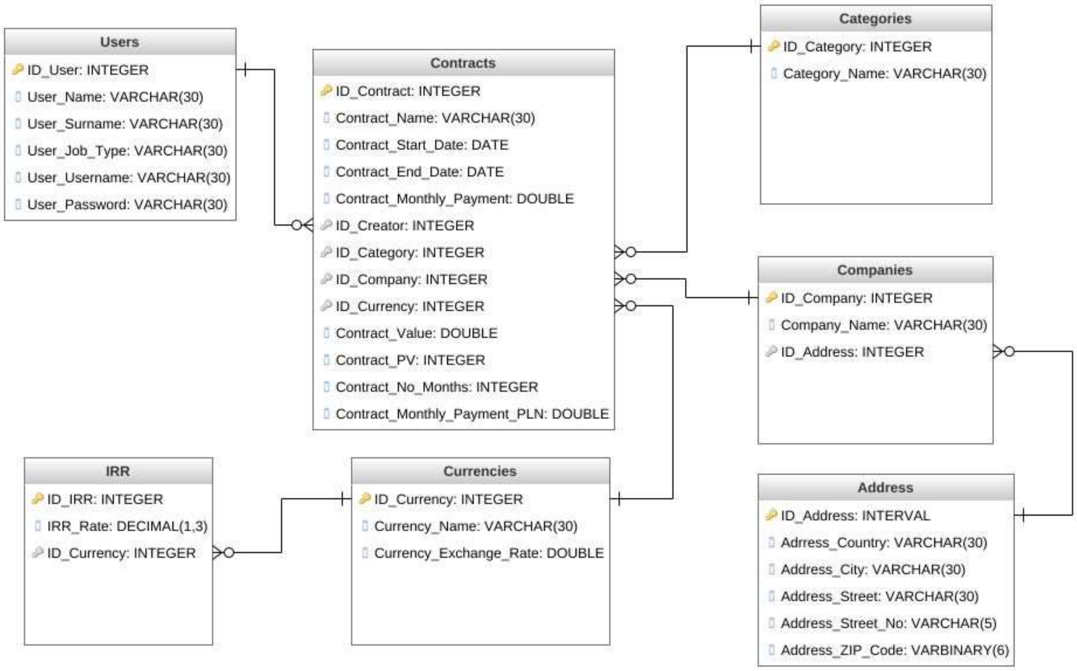
| | |
|---|---|
| Behbud Hamzayev | 436350 |
| Dawid Szyszko-Celiński | 443709 |

# INTRODUCTION

We wanted to make app interactive and user friendly. Of course, it is just a demo and should be only use in closed environment. App consists of panels: home, create contract, contract overview, admin panel (admin panel has 2 more panels Add user and Add contract), analyses, contact and login. When user opens the app for the first time there is a restriction with security rights. Before users log in, they have only access to home page, login page and contact page. If users try to go to different panels, they will get error message that they need to log in. When users log in there are messages which tells user if login was successful or not. If users are logged, they can access the create contract, contract overview and analyses panel additionally and make changes, analyse data there. Moreover, there are 2 types of users: admins and modifiers. Modifiers has only access to described panels whereas admin has additional rights to access admin panel in order to create new user or add category. When user is logged in but is not admin clicking admin panel will return an error message.

# Tables:

1. Users – table stores data about users, their name, surname, username and password. ID_User is primary key. Admins can update this table by going to "Admin Panel" and then "Add User" where we can provide all data and add the user to the database. There is no register button for the user as the app is assumed for business purposes where only admins can create and manage users.
2. Categories – table stores data about possible categories of contracts. Admins can create new category and update the table by going to "Admin panel" and then "Add Category". ID_Company is a primary key.
3. Companies – table stores data about Company Names. We assume there might be more than one company which uses same app to store data about contracts. There is foreign key ID_Address which connects table with Address table.
4. Address – table stores data about addresses of the companies.
5. Currencies – table stores names of the currencies and their exchange rates. ID_Currency is primary key.
6. IRR – table stores data about Internal Return Rate which is used to calculate PV of contract. IRR is based on the currency and it is connected by ID_currency foreign key. Primary key = ID_IRR
7. Contracts – main table storing data about the contracts. We can create new contract by "Create contract" panel and then look at them and filter by certain attributes in "Contract Overview". In "Analyses" tab in app we have visualisations of the data and different data is shown. Table stores basic data provided by user. User provides contract name, start date, end date, monthly payment, category, currency and company. Then python app calculates how many months are there between the start and end date and store it as No_Months. Then based on number of months Value is calculated as the formula = No_Months * Monthly_Payment. Contract PV is calculated by formula = monthly payment / (1 + IRR)^n when n is the number of month. Then for each month PV is calculated, summed and put into the database. Monthly payment PLN is basically monthly payment * exchange rate. The creator is provided in the template based on the currently logged user.


Users
ID_User: INTEGER
User_Name: VARCHAR(30)
User_Surname: VARCHAR(30)
User_Job_Type: VARCHAR(30)
User_Username: VARCHAR(30)
User_Password: VARCHAR(30)
Contracts
ID_Contract: INTEGER
Contract_Name: VARCHAR(30)
Contract_Start_Date: DATE
Contract_End_Date: DATE
Contract_Monthly_Payment: DOUBLE
ID_Creator: INTEGER
ID_Category: INTEGER
ID_Company: INTEGER
ID_Currency: INTEGER
Contract_Value: DOUBLE
Contract_PV: INTEGER
Contract_No_Months: INTEGER
Contract_Monthly_Payment_PLN: DOUBLE
Categories
ID_Category: INTEGER
Category_Name: VARCHAR(30)
Companies
ID_Company: INTEGER
Company_Name: VARCHAR(30)
ID_Address: INTEGER
IRR
ID_IRR: INTEGER
IRR_Rate: DECIMAL(1,3)
ID_Currency: INTEGER
Currencies
ID_Currency: INTEGER
Currency_Name: VARCHAR(30)
Currency_Exchange_Rate: DOUBLE
Address
ID_Address: INTERVAL
Adrress_Country: VARCHAR(30)
Address_City: VARCHAR(30)
Address_Street: VARCHAR(30)
Address_Street_No: VARCHAR(5)
Address_ZIP_Code: VARBINARY(6)

# SQL Queries:

### SQL Queries in Contract Overview:

1. Selecting distinct categories to make sure data is always in real time

2. Getting IRR_Rate to calculate PV

3. Getting Exchange rate to calculate exchanged values

4. Selecting currently logged user to put inside the database currently logged user

5. Inserting all the data to create contract after all calculations

### SQL Queries in Analyses:

1. Selecting data from Contracts table to show them in the table in template inside app

2. Selecting distinct categories to make pie chart and making sure it is updating in real time

**SQL Queries in Admin Panel:**

1. Inserting provided user data into the Users table

2. Selecting Job_Type from Users table to check if user has access to the panel

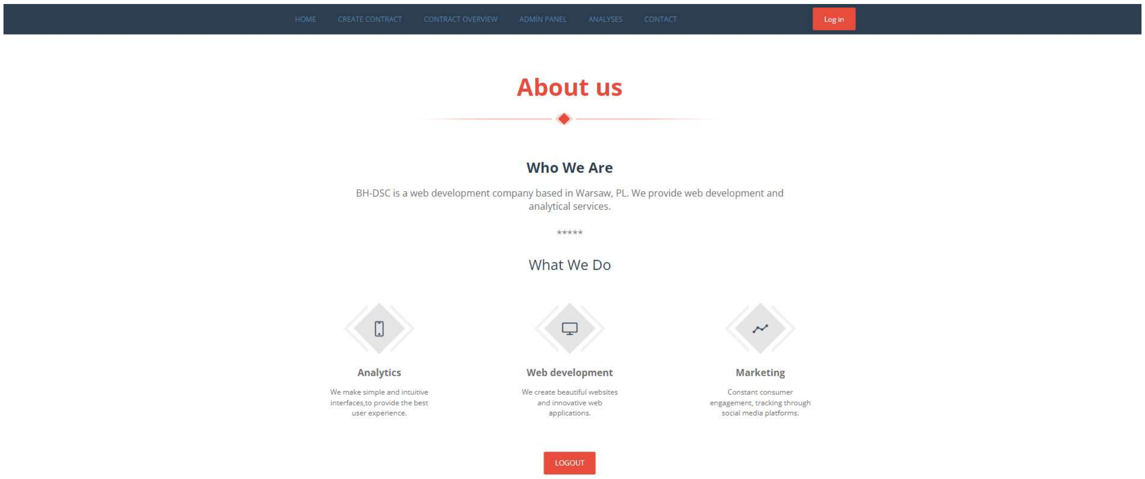
3. Inserting a new category with provided data

**SQL Queries in Login:**

1. Getting Users form Users table to check for the logged user

# Design:

1. **Home Page:** Our main page has a pleasant and intuitive interface in which you can access all the needed functions of our app.



About us

Who We Are

BH-DSC is a web development company based in Warsaw, PL. We provide web development and analytical services.

*****

What We Do

Analytics

We make simple and intuitive interfaces,to provide the best user experience.

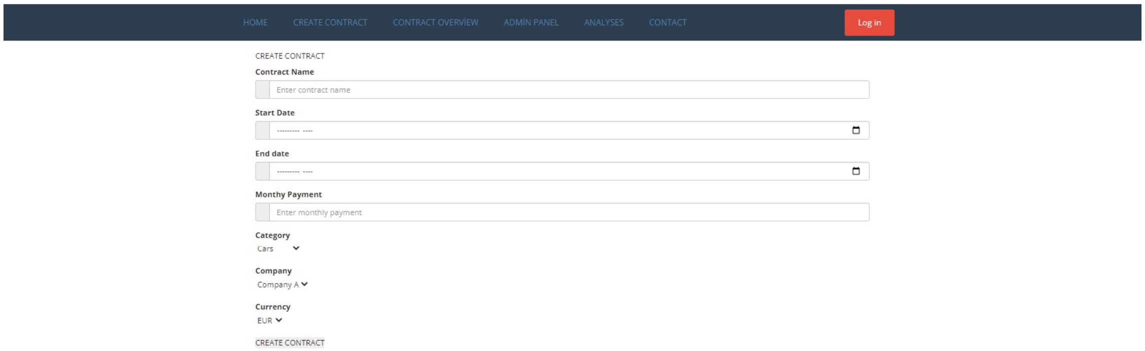
Web development

We create beautiful websites and innovative web applications.

Marketing

Constant consumer engagement, tracking through social media platforms.

2. Create Contract: Convenient page to crate easily new assigned contracts.



CREATE CONTRACT

Contract Name

Enter contract name

Start Date

--------- ----

End date

--------- ----

Monthy Payment

Enter monthly payment

Category

Cars

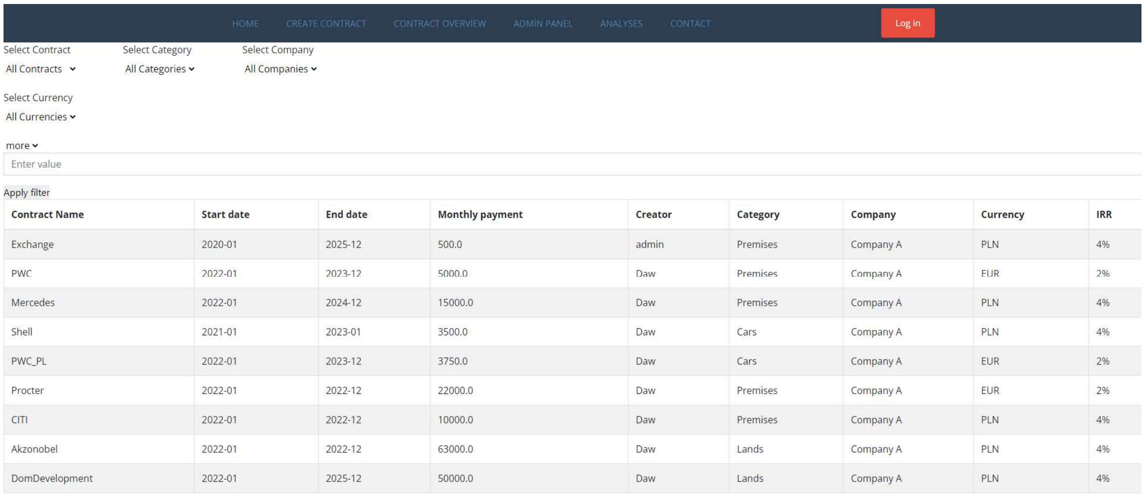
Company

Company A

Currency

EUR

## 3. Contract Overview:



Select Contract
All Contracts

Select Category
All Categories

Select Company
All Companies

Select Currency
All Currencies

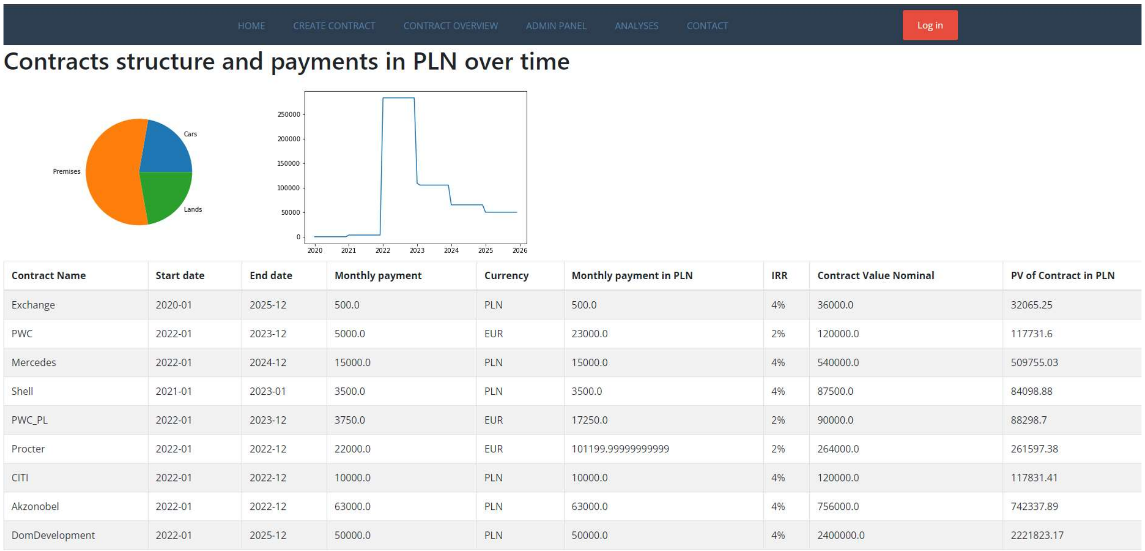
more
Enter value

Apply filter

| Contract Name | Start date | End date | Monthly payment | Creator | Category | Company | Currency | IRR |
|---|---|---|---|---|---|---|---|---|
| Exchange | 2020-01 | 2025-12 | 500.0 | admin | Premises | Company A | PLN | 4% |
| PWC | 2022-01 | 2023-12 | 5000.0 | Daw | Premises | Company A | EUR | 2% |
| Mercedes | 2022-01 | 2024-12 | 15000.0 | Daw | Premises | Company A | PLN | 4% |
| Shell | 2021-01 | 2023-01 | 3500.0 | Daw | Cars | Company A | PLN | 4% |
| PWC_PL | 2022-01 | 2023-12 | 3750.0 | Daw | Cars | Company A | EUR | 2% |
| Procter | 2022-01 | 2022-12 | 22000.0 | Daw | Premises | Company A | EUR | 2% |
| CITI | 2022-01 | 2022-12 | 10000.0 | Daw | Premises | Company A | PLN | 4% |
| Akzonobel | 2022-01 | 2022-12 | 63000.0 | Daw | Lands | Company A | PLN | 4% |
| DomDevelopment | 2022-01 | 2025-12 | 50000.0 | Daw | Lands | Company A | PLN | 4% |

Contract overview is meant to present basic info about contracts. We can see their info about name, start date, end date, monthly payment, category, creator, company, currency and IRR. Moreover, we have 5 filters. Unfortunately, user can only use one filter at the time, otherwise all contracts will be shown. Data can be filtered by Category, Company, Contract Name, Currency and monthly payment. Monthly payment can be used as a filter to show contracts with value bigger or smaller than provided threshold. By clicking apply filter data will be filtered. After creating new contract data is updated automatically and you can see new contract just after creation.

## 4. Analyses:

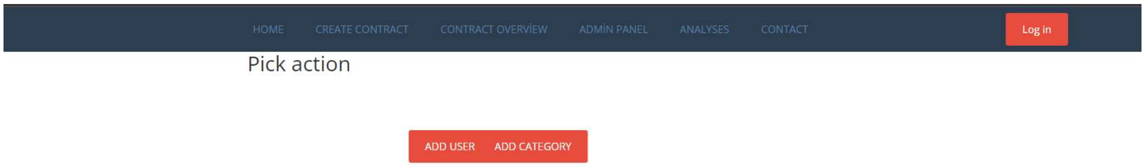


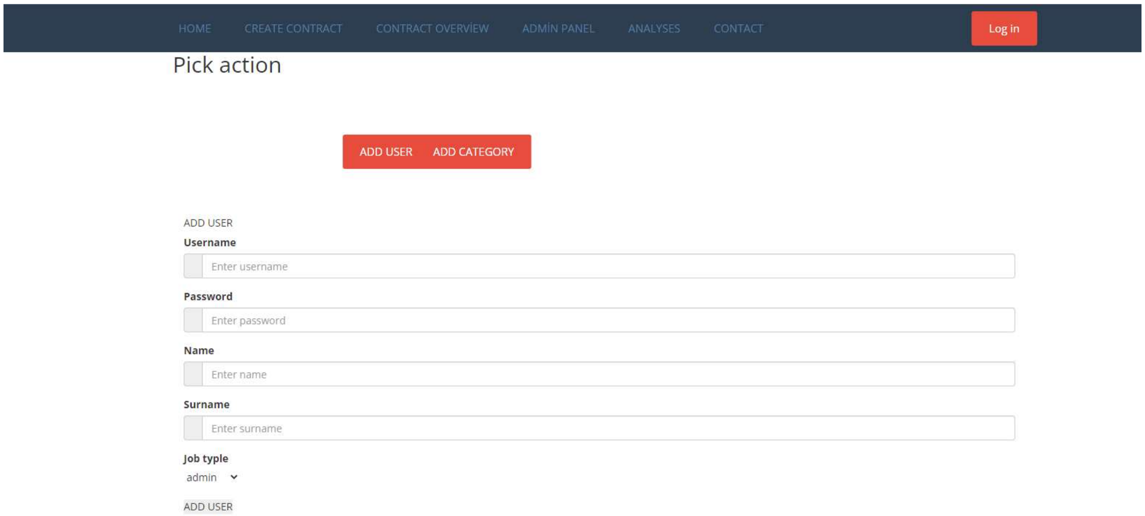
Contracts structure and payments in PLN over time

| Contract Name | Start date | End date | Monthly payment | Currency | Monthly payment in PLN | IRR | Contract Value Nominal | PV of Contract in PLN |
|---|---|---|---|---|---|---|---|---|
| Exchange | 2020-01 | 2025-12 | 500.0 | PLN | 500.0 | 4% | 36000.0 | 32065.25 |
| PWC | 2022-01 | 2023-12 | 5000.0 | EUR | 23000.0 | 2% | 120000.0 | 117731.6 |
| Mercedes | 2022-01 | 2024-12 | 15000.0 | PLN | 15000.0 | 4% | 540000.0 | 509755.03 |
| Shell | 2021-01 | 2023-01 | 3500.0 | PLN | 3500.0 | 4% | 87500.0 | 84098.88 |
| PWC_PL | 2022-01 | 2023-12 | 3750.0 | EUR | 17250.0 | 2% | 90000.0 | 88298.7 |
| Procter | 2022-01 | 2022-12 | 22000.0 | EUR | 101199.99999999999 | 2% | 264000.0 | 261597.38 |
| CITI | 2022-01 | 2022-12 | 10000.0 | PLN | 10000.0 | 4% | 120000.0 | 117831.41 |
| Akzonobel | 2022-01 | 2022-12 | 63000.0 | PLN | 63000.0 | 4% | 756000.0 | 742337.89 |
| DomDevelopment | 2022-01 | 2025-12 | 50000.0 | PLN | 50000.0 | 4% | 2400000.0 | 2221823.17 |

Analyses shows different data about contracts. First, we can see the pie chart based on contracts category which shows structure of all contracts. Based on that we can get info what is most created type of contract. Moreover, we have plot which shows payments over time in each month after exchange for PLN. Based on that plot we can see in which month how much will we have to pay for contracts in PLN and see how will it change over time. Plots and table below is updating automatically. If admin add new category it will appear in the pie chart. The table shows data similar to the contract overview but additionally we have payments after exchange to PLN. Value of whole contract calculated as monthly payment * month of contract. There is also Present Value of the contract calculated based on IRR which is based on currency.

5. **Admin Panel:** Admin panel can be accessed only users with admin status. Furthermore, admins through this page can create users and add new additional categories. The registration of new users in this way gave our app more reliability and security measures.





ADD USER

Username

Enter username

Password

Enter password

Name

Enter name

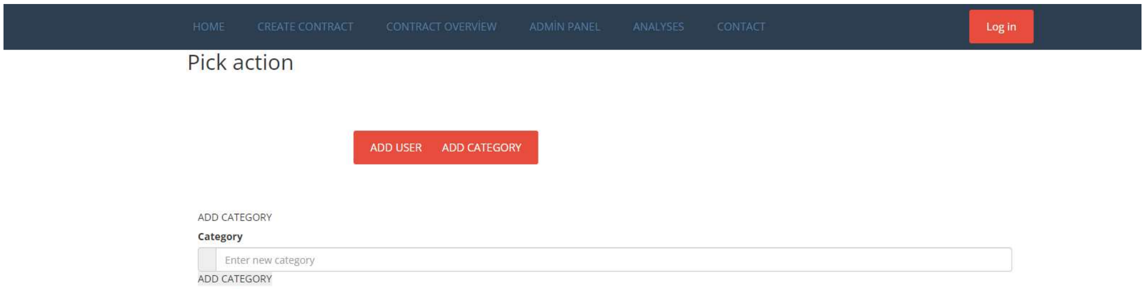
Surname

Enter surname

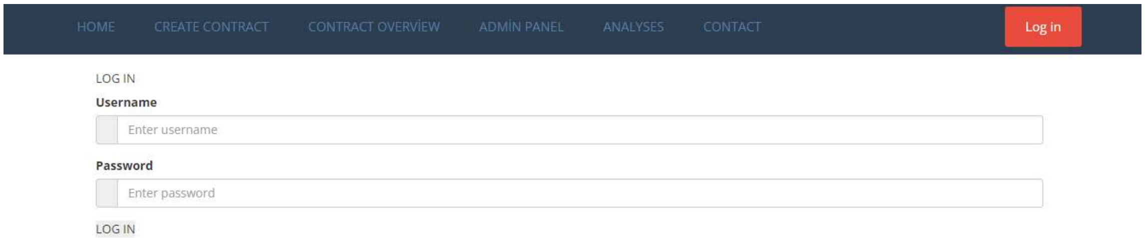
Job typle

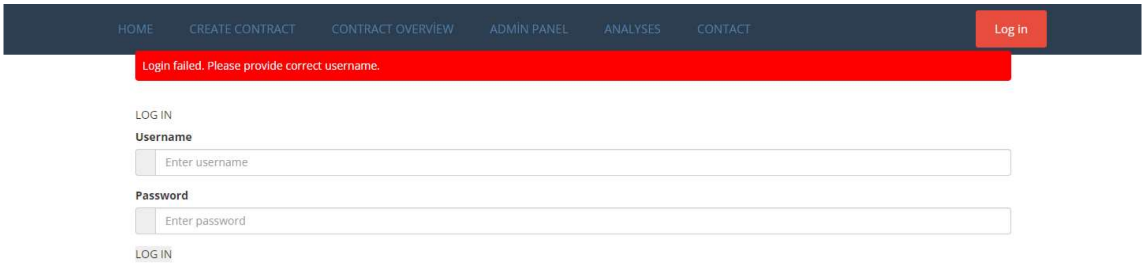
admin

ADD USER



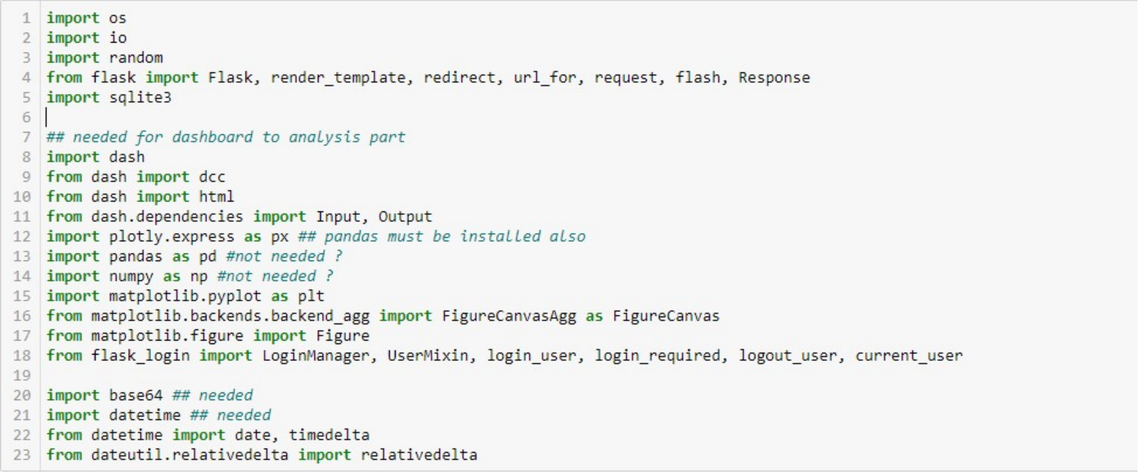
6. **Log in:** On the login page, user-input data has to be verified through the "Users" table in the Database. If inputted date fails verification the will pop a red error message as shown below in screenshots.





Login failed. Please provide correct username.

## External Python Packages:



```python
import os
import io
import random
from flask import Flask, render_template, redirect, url_for, request, flash, Response
import sqlite3

## needed for dashboard to analysis part
import dash
from dash import dcc
from dash import html
from dash.dependencies import Input, Output
import plotly.express as px ## pandas must be installed also
import pandas as pd #not needed ?
import numpy as np #not needed ?
import matplotlib.pyplot as plt
from matplotlib.backends.backend_agg import FigureCanvasAgg as FigureCanvas
from matplotlib.figure import Figure
from flask_login import LoginManager, UserMixin, login_user, login_required, logout_user, current_user

import base64 ## needed
import datetime ## needed
from datetime import date, timedelta
from dateutil.relativedelta import relativedelta
```

## Distribution of Work:

1) Idea/short description – Dawid Szyszko-Celiński (DSC)
2) Templates – Behbud Hamzayev (BH)
3) Database Scheme (DSC)
4) Register, login page (BH)
5) Creating Database (DSC)
6) Creating tabs and visual side (BH)
7) Creating DB connection and dropdown functionality (DSC)
8) SQL statements and functionality to grab data from templates ( both – we will divide all SQL functionalities by half)
9) Report and presentation (both – doing parts based on what we were doing in the project)

**References:**

Our templates were based on the templates from https://bootsnipp.com/. Of course, most of them were modified to needs of the project.