Coding and Implementation

The code for the project is available online on GitHub
link: https://github.com/P-Manoj-Kumar/ip-final-project

loader.py

```
 import pandas as pd
import streamlit as st
import country_converter as coco
from sqlalchemy import create_engine


def load_data() -> pd.DataFrame:

    try:
        connection = create_engine(
            f"mysql+pymysql://root:{st.secrets.root_password}@localhost:3306/data"
        ).connect()

        df = pd.read_sql_table(table_name="salaries", con=connection, schema="data")

    finally:
        connection.close()

    return clean_df(df)


def clean_df(df: pd.DataFrame) -> pd.DataFrame:
    remote_ratio_map = {
        0: "No remote work",
        50: "Partially remote",
        100: "Fully remote",
    }

    experience_level_map = {
        "EN": "Junior",
        "MI": "Intermediate",
        "SE": "Expert",
        "EX": "Director",
    }

    employment_type_map = {
        "PT": "Part-time",
        "FT": "Full-time",
        "CT": "Contract",
        "FL": "Freelance",
    }

    company_size_map = {
        "S": "Small",
        "M": "Medium",
        "L": "Large",
    }

    df: pd.DataFrame = (
        df
```

```
        .astype(
            {
                "work_year": "category",
                "experience_level": pd.CategoricalDtype(
                    experience_level_map.keys(), ordered=True
                ),
                "employment_type": pd.CategoricalDtype(
                    employment_type_map.keys(), ordered=True
                ),
                "company_size": pd.CategoricalDtype(company_size_map.keys(), ordered=True),
                "remote_ratio": pd.CategoricalDtype(remote_ratio_map.keys(), ordered=True),
            }
        )
        .drop(columns=["salary", "salary_currency"])
        .replace(
            {
                "remote_ratio": remote_ratio_map,
                "experience_level": experience_level_map,
                "employment_type": employment_type_map,
                "company_size": company_size_map,
                "employee_residence": dict(
                    zip(
                        df.employee_residence.unique(),
                        coco.convert(df.employee_residence.unique(), to="name_short"),
                    )
                ),
                "company_location": dict(
                    zip(
                        df.company_location.unique(),
                        coco.convert(df.company_location.unique(), to="name_short"),
                    )
                ),
            },
        )
        .assign(
            working_overseas=lambda x: x.employee_residence != x.company_location
        )

    )
    return df
```

🏠_Home.py

```
import streamlit as st

from utils.loader import load_data


if "df" not in st.session_state:
    st.session_state.df = st.cache(load_data)()


st.image('./images/cover.jpeg')

st.markdown(
f"""
<h3 align="center">
Informatics Practices Investigatory Project (2022-23)
</h3>
---
```

```
---
<h2 align="center">
Web-based Interactive Dashboard for
</h2>

<h1 align="center">
Job Salaries in Data Science
</h1>

---

### About the Dataset:

**Source**: https://salaries.ai-jobs.net/download/

**Shape**: `{st.session_state.df.shape[0]}` _rows_ x `{st.session_state.df.shape[1]}` _columns_

---

### Columns:

1. **work_year**: The year the salary was paid.
1. **experience_level**: The experience level in the job during the year with the following possible v
    - Junior
    - Intermediate
    - Expert
    - Director

1. **employment_type**: The type of employment for the role:
    - Part-time
    - Full-time
    - Contract
    - Freelance

1. **job_title**: The role worked in during the year.

1. **salary_in_usd**: The total gross salary amount paid in USD.

1. **employee_residence**: Employee's primary country of residence in during the work year

1. **remote_ratio**: The overall amount of work done remotely, possible values are as follows:
    - No remote work (less than 20%)
    - Partially remote
    - Fully remote (more than 80%)

1. **company_location**: The country of the employer's main office or contracting branch

1. **company_size**: The average number of people that worked for the company during the year.
    - Small (less than 50 employees)
    - Medium (50 to 250 employees)
    - Large (more than 250 employees)
""",
unsafe_allow_html=True
)

st.dataframe(st.session_state.df.sample(n=5, random_state=42))
```