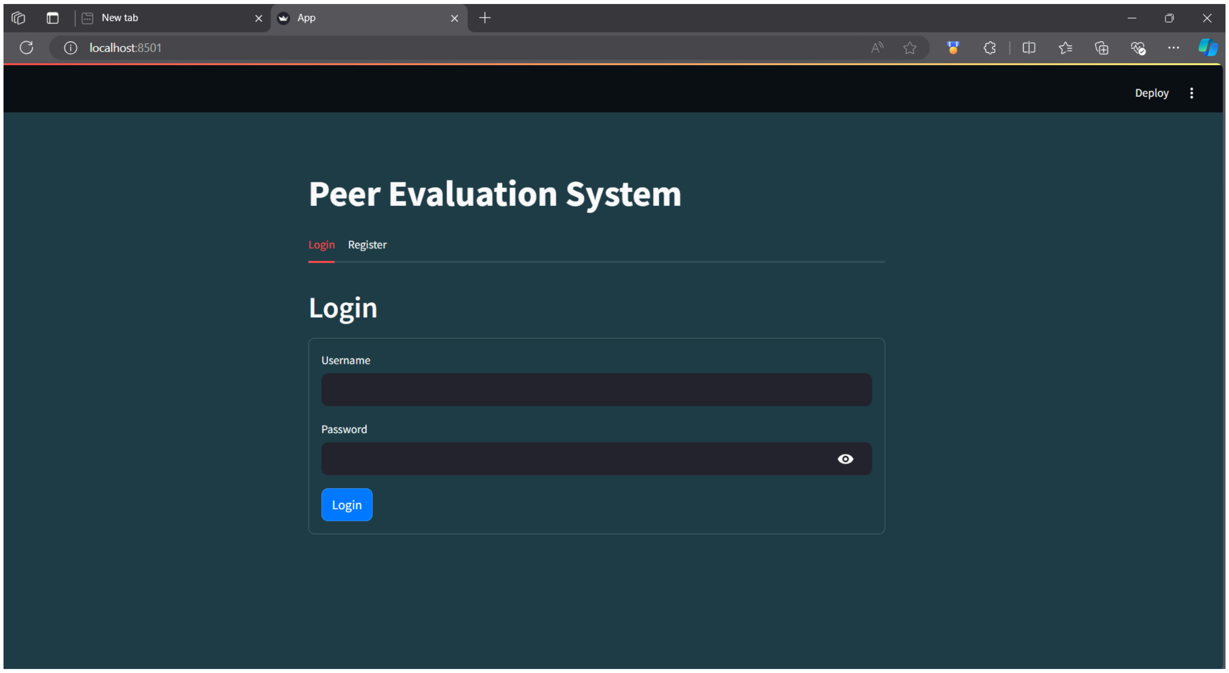
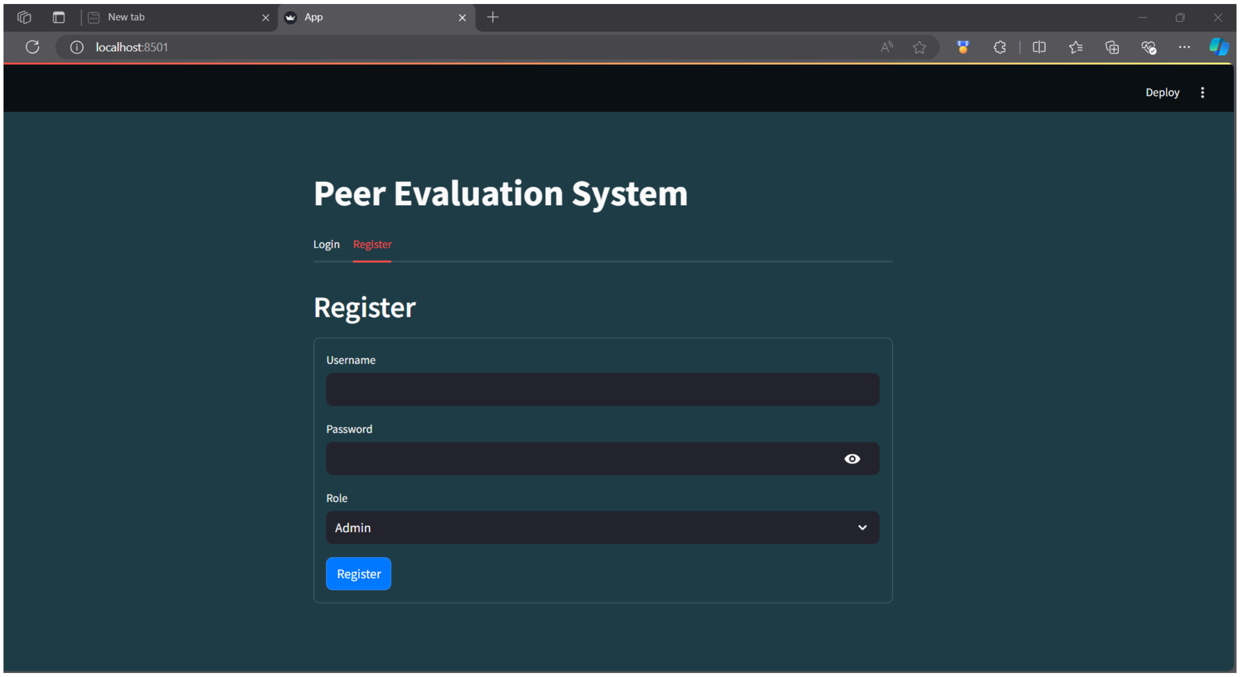
Peer Evaluation System UI/UX

Sample Screenshots of the proposed UI/UX design: -

- The below images shows the Login window for the proposed system: -
  Login Page: -



Registration Page: -



Role
Admin

Code: -

```
import streamlit as st
import gspread
from oauth2client.service_account import ServiceAccountCredentials

# Google Sheets setup
SCOPE = ["https://spreadsheets.google.com/feeds",
"https://www.googleapis.com/auth/drive"]
CREDENTIALS_FILE = "D:/ROHIT IIT/Peer
Evaluation/peer-evaluation-sem1-e2fcf8b5fc27.json"
SHEET_NAME = "UserRoles"

# Initialize connection to Google Sheets
def connect_to_google_sheets():
    creds = ServiceAccountCredentials.from_json_keyfile_name(CREDENTIALS_FILE,
SCOPE)
    client = gspread.authorize(creds)
    sheet = client.open(SHEET_NAME).sheet1
    return sheet

# Fetch users from Google Sheets
def get_users_from_sheets():
    sheet = connect_to_google_sheets()
    records = sheet.get_all_records()
    return records

# Add new user to Google Sheets
def register_user(username, password, role):
    sheet = connect_to_google_sheets()
    new_user = [username, password, role]
    sheet.append_row(new_user)

# Verify user credentials
def login(username, password, users):
    for user in users:
        if user['username'] == username and user['password'] == password:
            st.session_state["login_status"] = True
            st.session_state["role"] = user["role"]
            st.session_state["username"] = username
            st.session_state["page"] = "dashboard"
            st.session_state["message"] = None
            return
    st.session_state["message"] = "Incorrect username or password"

# Logout function
def logout():
    st.session_state["login_status"] = False
    st.session_state["role"] = None
    st.session_state["username"] = None
    st.session_state["page"] = "login"
    st.session_state["message"] = "Logged out successfully"

# Role-based content
def admin_dashboard():
    st.title("Admin Dashboard")
    st.write("Admins can manage everything.")

def teacher_dashboard():
    st.title("Teacher Dashboard")
    st.write("Teachers can manage quizzes and view results.")


def ta_dashboard():
    st.title("TA Dashboard")
    st.write("TAs can assist in managing student assessments.")

def student_dashboard():
    st.title("Student Dashboard")
    st.write("Students can evaluate peers and view feedback.")

# Main Streamlit app
def main():
    # Initialize session state variables if not present
    if "login_status" not in st.session_state:
        st.session_state["login_status"] = False
    if "role" not in st.session_state:
        st.session_state["role"] = None
    if "username" not in st.session_state:
        st.session_state["username"] = None
```

```
if "page" not in st.session_state:
    st.session_state["page"] = "login"
if "message" not in st.session_state:
    st.session_state["message"] = None

# Set background color and input field styling using HTML
st.markdown(
    """
    <style>
    .stApp {
        background-color: #1f3f49;  /* Light blue background */
    }
    .stTextInput>div>input, .stPasswordInput>div>input {
        background-color: white;  /* White background for text and password inputs */
        color: black;  /* Text color for input fields */
    }
    .stButton>button {
        background-color: #007bff;  /* Optional: Style buttons with a color */
        color: white;
    }
    </style>
    """,
    unsafe_allow_html=True
)

# Page routing based on session state
if st.session_state["page"] == "login":
    st.title("Peer Evaluation System")

    # Tabs for Login and Registration
    tab1, tab2 = st.tabs(["Login", "Register"])

    with tab1:
        st.header("Login")

        with st.form(key='login_form'):
            username = st.text_input("Username")
            password = st.text_input("Password", type="password")
            submit_button = st.form_submit_button("Login")
```

```
                if submit_button:
                    users = get_users_from_sheets()
                    login(username, password, users)
                    if st.session_state["login_status"]:
                        st.rerun()

        with tab2:
            st.header("Register")

            with st.form(key='register_form'):
                reg_username = st.text_input("Username", key='reg_username')
                reg_password = st.text_input("Password", type="password",
key='reg_password')
                role = st.selectbox("Role", ["Admin", "Teacher", "TA", "Student"])
                register_button = st.form_submit_button("Register")

                if register_button:
                    if not reg_username.endswith("@iitrpr.ac.in"):
                        st.error("Username must end with @iitrpr.ac.in")
                    else:
                        users = get_users_from_sheets()
                        if any(user['username'] == reg_username for user in users):
                            st.error("Username already exists")
                        else:
                            register_user(reg_username, reg_password, role)
                            st.success("User registered successfully")
                            # Redirect to the login page
                            st.session_state["page"] = "login"
                            st.rerun()

    elif st.session_state["page"] == "dashboard":
        if st.session_state["role"] == "Admin":
            admin_dashboard()
        elif st.session_state["role"] == "Teacher":
            teacher_dashboard()
        elif st.session_state["role"] == "TA":
            ta_dashboard()
        elif st.session_state["role"] == "Student":
            student_dashboard()
        # Logout button
        if st.button("Logout"):
            logout()
            st.rerun()

if __name__ == "__main__":
    main()
```