

Sprint 1- User Login Page **Code:**

```
import pandas as pd
import plotly.express as px
import streamlit as st
import streamlit_authenticator as stauth
import database as db

st.set_page_config(page_title="Crop Yield Dashboard",
                    page_icon=":bar_chart:", layout="wide")
placeholder = st.empty()
users = db.fetch_all_users()
usernames = [user["key"] for user in users]
names = [user["name"] for user in users]
hashed_passwords = [user["password"] for user in users]
authenticator = stauth.Authenticate(names, usernames, hashed_passwords,
                                     "sales_dashboard", "abcdef", cookie_expiry_days=30)

name, authentication_status, username = authenticator.login("Login", "main")

if authentication_status == False:
    st.error("Username/password is incorrect")
elif authentication_status == None:
    st.warning("Please enter your username and password")
elif authentication_status:
    #enter the dashboard page
```

```
import os
import database as db
from deta import Deta # pip install deta
from dotenv import load_dotenv # pip install python-dotenv
# Load the environment variables
load_dotenv(".env")
```

```

#DETA_KEY = os.getenv("DETA_KEY")
DETA_KEY="d00zeamk_EPLsmaf4njyYDRJZVBUL4ayqf3t9B26F"

# Initialize with a project key
deta = Deta(DETA_KEY)

# This is how to create/connect a database db
db = deta.Base("crop")

def insert_user(username, name, password):
    """Returns the user on a successful user creation, otherwise raises
    and error"""
    return db.put({"key": username, "name": name,
"password":
password})

def fetch_all_users():
    """Returns a dict of all users"""
    res = db.fetch()
    return res.items

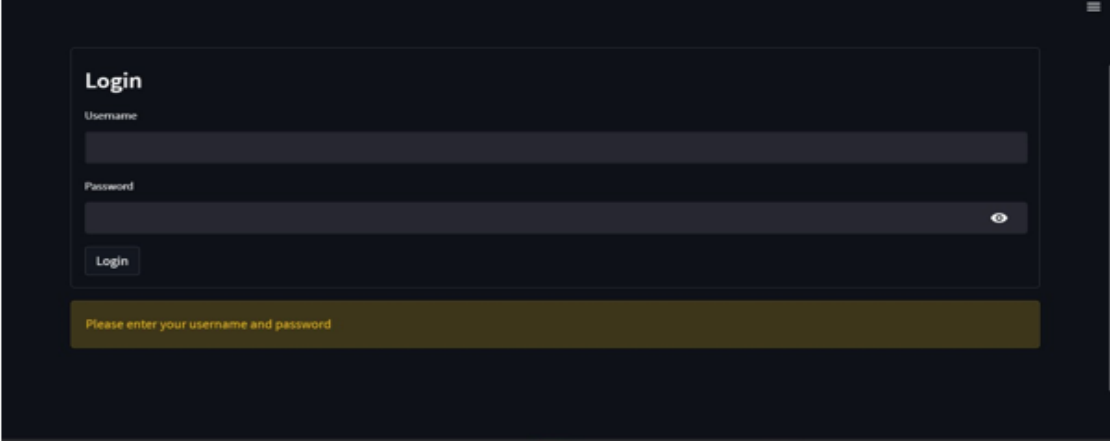
def get_user(username):
    """If not found, the function will return None"""
    return db.get(username)

def update_user(username, updates):
    """If the item is updated, returns None. Otherwise, an exception is
    raised"""
    return db.update(updates, username)

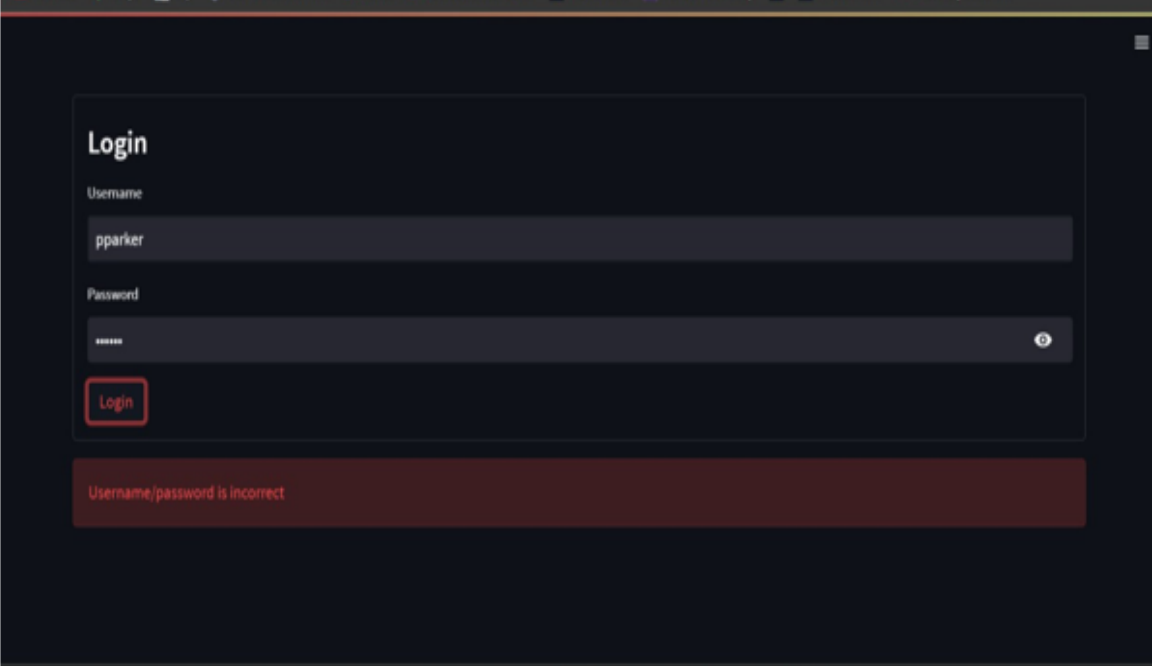
```

```
def delete_user(username):  
    """Always returns None, even if the key does not exist"""  
    return db.delete(username)
```

Output:



A screenshot of a web application's login page. The page has a dark blue background. At the top left, there is a 'Login' section with a title 'Login' in white. Below the title are two input fields: 'Username' and 'Password'. The 'Username' field is empty, and the 'Password' field is also empty with a small eye icon to its right. Below the input fields is a 'Login' button. At the bottom of the login section, there is a yellow message box that says 'Please enter your username and password'.



A screenshot of the same web application's login page. The 'Username' field now contains the text 'pparker' and the 'Password' field contains several dots. The 'Login' button is highlighted with a red border. Below the login section, there is a red message box that says 'Username/password is incorrect'.