

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")
if authentication_status == None:
    st.warning("Please enter your username and password")
if authentication_status:
```

```
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 = 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:

