# Playwright Dynamic Login Using .env + Enum

```
===========================
1) Improved .env Structure
===========================

QA_USER_FA01_NAME=FA01-neo3694
QA_USER_FA01_PASSWORD=FA01@123

QA_USER_SE12558_NAME=se12558
QA_USER_SE12558_PASSWORD=Pwd@567

QA_USER_TEST0987_NAME=Test0987
QA_USER_TEST0987_PASSWORD=Pass@987

----------------------------
2) Enum for User Types
----------------------------

export enum UserType {
  FA01 = "FA01",
  SE12558 = "SE12558",
  TEST0987 = "TEST0987"
}

----------------------------------------------
3) Utility to Read Credentials from .env File
----------------------------------------------

import * as dotenv from "dotenv";
dotenv.config();
import { UserType } from "./UserType";

export class UserCredentials {
  static getCredentials(user: UserType) {
    const username = process.env[`QA_USER_${user}_NAME`];
    const password = process.env[`QA_USER_${user}_PASSWORD`];

    if (!username || !password) {
      throw new Error(`Credentials not found for user: ${user}`);
    }
    return { username, password };
  }
}

----------------------
4) Login Page Usage
----------------------

import { Page } from "@playwright/test";
import { UserType } from "../enums/UserType";
import { UserCredentials } from "../utils/UserCredentials";

export class LoginPage {
  constructor(private page: Page) {}
  async loginAs(user: UserType) {
    const { username, password } = UserCredentials.getCredentials(user);
    await this.page.fill("#username", username);
    await this.page.fill("#password", password);
    await this.page.click("#loginButton");
  }
}

-----------------------
5) Test Case Example
-----------------------

import { test } from "@playwright/test";
import { LoginPage } from "../pages/LoginPage";
import { UserType } from "../enums/UserType";

test("Login with FA01 user", async ({ page }) => {
  const login = new LoginPage(page);
  await login.loginAs(UserType.FA01);
});
```

```
=======================================
■ Benefits of This Approach
=======================================

✔ No hard-coded credentials
✔ Dynamic selection using Enum
✔ Easy to add new users
✔ Safe through .env file

---------------------------------------
■ Optional Upgrade (Available on Request)
---------------------------------------
■  Password Encryption
■  Multi-environment Support (DEV/QA/PROD)
```