import { test, Locator, Page } from '@playwright/test'

export default class PharmacyPage {

  readonly page: Page;

  private pharmacy: Locator;

  private order: Locator;

  private export: Locator;

  downloadPath: string | undefined;

  constructor(page: Page) {

    this.page = page;

    this.pharmacy = page.getByRole('link', { name: 'Pharmacy ' });

    this.order = page.locator(`//a[@href="#/Pharmacy/Order"]`).nth(1);

    this.export = page.getByTitle(`Export To Excel`);

  }

  /\*\*

   \* This method verifies the functionality of exporting the order section data. It first waits for a brief timeout

   \* to ensure that the page elements are fully loaded. Then, it navigates through the pharmacy section and selects

   \* the order option once it's visible. After that, the method waits for the file download to begin by listening for

   \* the 'download' event. Once the export button is clicked, it waits for the download to complete and saves the

   \* file to the specified download directory. The file path of the downloaded file is stored in the `downloadPath`

   \* variable for further use or validation. This method helps in validating that the export functionality works as

   \* expected and the file is successfully downloaded.

   \*/

  /\*\*

   \* @Test10 Verify to export the order section data

   \*

   \* 1. Navigate to https://healthapp.yaksha.com/Home/Index#/Pharmacy/Dashboard

   \* 2. Click on order section

   \* 3. Click on "Export" button

   \*

   \*/

  async verifyExportOrderSectionData() {

    const counter = await this.page.locator("//a[text()='X']");

    if (counter) {

      await counter.click();

    }

    await this.pharmacy.click()

    await this.order.click()

    const [download] = await Promise.all([

      this.page.waitForEvent('download'),

      await this.export.click()

    ]);

    const path = await download.path();

    this.downloadPath = path;

    return path;

  }

}