**ID-Medical**

**NHSP Portal Python Parser**

**Document History**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Reviewed By** | **Approved By** | **Description** |
| 29th Jun 2023 | Initial draft version | Kapil Patil |  |  | Initial Draft version send for review |

Table of Contents

[**Introduction** 3](#_Toc138927100)

[Background 3](#_Toc138927101)

[Scope 3](#_Toc138927102)

[Overview 3](#_Toc138927103)

[**Functional Requirements Summary** 3](#_Toc138927104)

[System Considerations 3](#_Toc138927105)

[Assumptions: 3](#_Toc138927106)

[System Environment: 4](#_Toc138927107)

[Solution 4](#_Toc138927108)

[Functional Flow: 4](#_Toc138927109)

[Technical Solution Design 4](#_Toc138927110)

[Technical/Functional Flow 5](#_Toc138927111)

[**Deployment Strategy** 5](#_Toc138927112)

[**Effort Estimations** 5](#_Toc138927113)

# **Introduction**

## Background

ID-Medical was looking for the solution which can parse the NHSP web portal using Python. After analyzing the requirements, we have created the solution approach which allows to parse the web portal and scrape the required requirement details and save that in the excel file. Other developer team can use that excel data for the further processing.

## Scope

Scope of work includes:

* Low Level Design
* System design and development

## Overview

ID-Medical would like to develop a Python scraping functionality for the following web portal:

* NHSP Portal (<https://bank.nhsp.uk/ourbank>)

# **Functional Requirements Summary**

This Python functionality will parse and scrape the web portal for login and to fetch the unfilled requirement details into the excel file. This Python function will be schedule using the Windows Task Scheduler for the required frequency.

## System Considerations

### Assumptions:

* Any changes in web portal HTML structure will lead to additional efforts
* This will not involve any database related activities

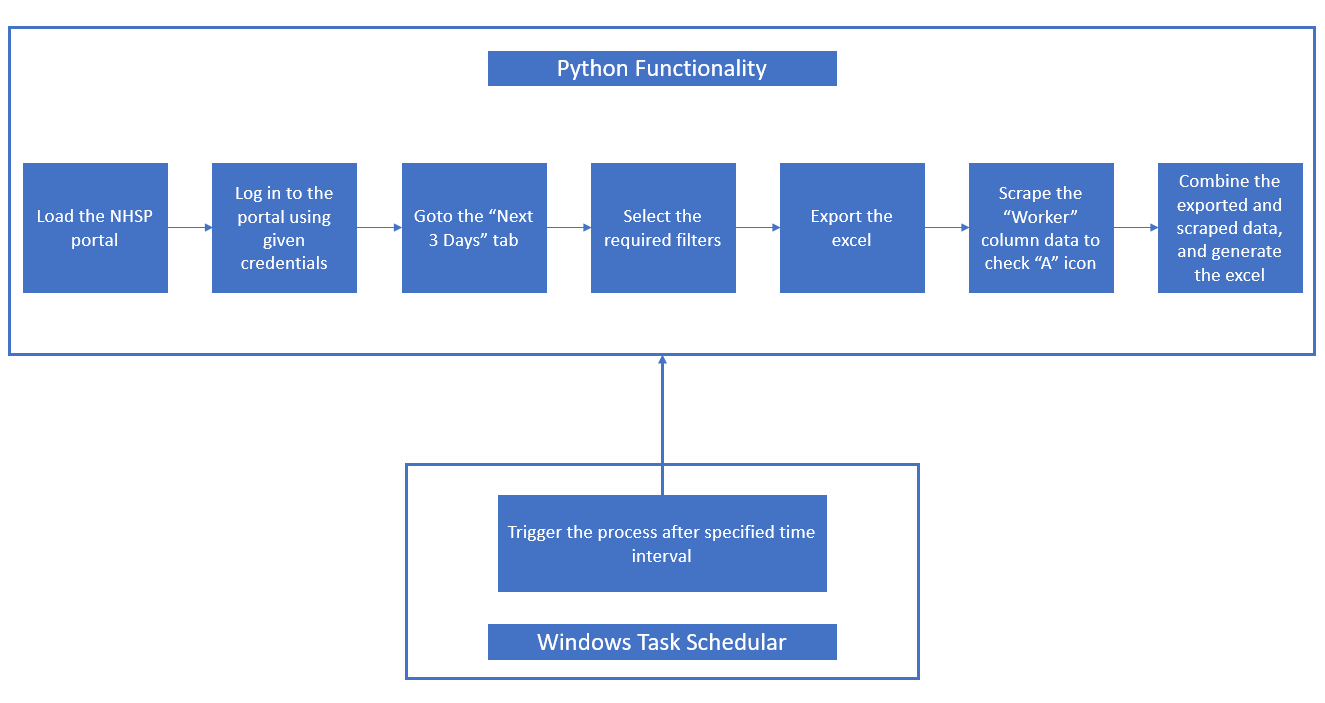
### System Environment:

|  |  |  |
| --- | --- | --- |
| **Area** | **Technology** | **Remarks** |
| **Core Language/ Framework** | Python (Selenium) | Most popular but we need to take call on python version to use |

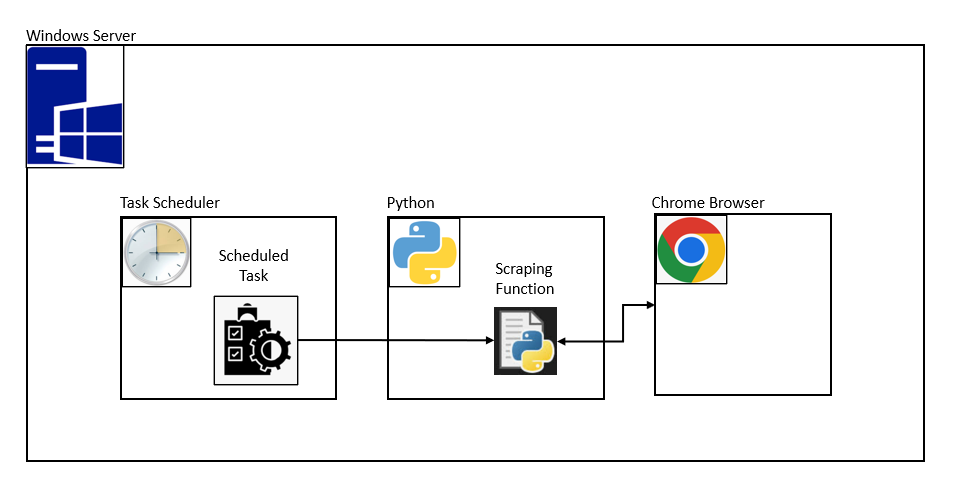
## 

## Solution

### Functional Flow:



### Technical Solution Design



### Technical/Functional Flow

1. Windows task scheduler will trigger the Python scraping functionality.
2. Python will load the NHSP portal login page using the selenium.
3. Log into the portal using the provided credentials.
4. Parse the page after login and click on the “Next 3 Days” tab.
5. Select the required filtering options.
6. Export the excel file using the “Export” button provided.
7. Parse the table by rows and scrape the “Worker” column to check the “A” icon.
8. If “A” icon presents for that row then “Worker” column will be set to value 1 in shift requirement excel. Else “Worker” column will be set to value 0 in shift requirement excel.
9. After processing all the rows from exported excel and table, generate the scraped shift requirements excel.

# **Deployment Strategy**

1. Install the required version of Python on the deployment server.
2. Update the environment varibles.
3. Install/update the Chrome browser (if not installed/updated).
4. Add the developed code and create the python virtual environment and install the required dependencies.
5. Create the required folder structure to store the exported and generated excel files.
6. Schedule the task in “Task Scheduler” to trigger the Python function.

# **Effort Estimations**

|  |  |
| --- | --- |
| **Activities** | **Efforts (Hrs)** |
| Backend development | 40 |
| Deployment & DevOps | 10 |
| QA and Testing | 8 |
| Total | 58 |