**Identify Automation Candidates**

**Task01\_HomeTask\_ROI**

Taking Example of Amazon.in E-Commerce project:

A general approach to fill in the data based on the context of Amazon's project, assuming we're dealing with a typical large-scale e-commerce platform.

**1. Should we implement automation on the project? Why?**

**Yes**, implementing automation on a project like Amazon is highly recommended due to:

* The large number of test cases required for various features (product searches, payments, user authentication, etc.).
* Frequent updates and releases, requiring regression testing.
* Automation improves efficiency by reducing manual effort and ensuring consistency across test executions.

**2. What can we automate? Why?**

We can automate the following:

* **Regression Testing**: Ensures that new changes do not break existing functionality.
* **API Testing**: Automating backend APIs to check if different services interact correctly.
* **Critical User Flows**: Automating checkout flows, product searches, user login, etc., since these are high-risk areas that need frequent testing.
* **Cross-browser Testing**: Ensuring that the platform works across different browsers and devices is easier with automation. Automating these areas will save time, increase accuracy, and reduce repetitive manual tasks.

**3. ROI Calculation**

Using the ROI formula, we can fill in the table based on the given example:

**Manual testing cost**:  
Assume 30 man-hours per week over 5 years = 7800 man-hours.

**Automated tests development cost**:

* Framework setup: 120 man-hours
* Scenario development: 60 man-hours per week over 0.6 years = 1872 man-hours.

**Execution result analysis**:  
10 man-hours per week over 5 years = 2600 man-hours.

**ROI Calculation:**

**A black numbers with white text

Description automatically generated with medium confidence**

The filled table based on an assumed 5-year automation plan for Amazon:

| **Attributes** | **Year1** | **Year2** | **Year3** | **Year4** | **Year5** |
| --- | --- | --- | --- | --- | --- |
| Total No of Test Cases | 1000 | 1200 | 1500 | 1800 | 2000 |
| Total No of Automated TCs | 100 | 300 | 500 | 700 | 1000 |
| Total No of Testing cycles | 12 | 12 | 12 | 12 | 12 |
| Time req to execute a TC manually (mins) | 10 | 10 | 10 | 10 | 10 |
| Time req to execute a TC via script (mins) | 2 | 2 | 2 | 2 | 2 |
| Time req to develop script for a TC (hrs) | 3 | 3 | 3 | 3 | 3 |
| Cost of 1 resource in ($) | 30 | 30 | 30 | 30 | 30 |
| Total cost of tool in ($) | 5000 | 0 | 0 | 0 | 0 |
| Total Manual Effort in (hrs) | 1800 | 1800 | 1800 | 1800 | 1800 |
| Total Automation Effort in (hrs) | 600 | 800 | 1000 | 1000 | 1100 |
| Total Cost of Manual Testing in ($) | 54000 | 54000 | 54000 | 54000 | 54000 |
| Total Cost of Automation Testing | 22000 | 15000 | 12000 | 10000 | 8000 |
| Total Cumulative Saving | 32000 | 41000 | 50000 | 59000 | 66000 |
| ROI | 31% | 49% | 64% | 79% | 88% |

**Key Assumptions:**

* **Total No of Test Cases**: Increased over time as features grow.
* **Total No of Automated Test Cases**: Starts small and scales as more are automated each year.
* **Cost of Manual Testing**: Remains constant, but automation saves cumulative effort over time.
* **ROI**: Gradually increases as automation effort reduces testing costs over the years.