

## Q1. A)

A development team of 10 members is given 3 months (90 days) to develop a **Maha-Khumb Mobile App**.

### Breaking Down the Work into User Stories

Key Features:

- ✓ **User Registration & Ticket Booking**
- ✓ **Live Event Streaming**
- ✓ **Navigation & Crowd Management**
- ✓ **Emergency Alerts & Notifications**

The team assigns **Story Points (SP)** using Planning Poker.

User Story	SP
User Registration	5
Ticket Booking	13
Live Streaming	21
Navigation	8
Emergency Alerts	5

Total Story Points=5+13+21+8+5=52

### Calculate Team Velocity

Velocity is the **average number of story points completed per Sprint**.

Let's assume the team follows **2-week Sprints** and their velocity is **20 Story Points per Sprint**.

Number of Sprints=Total Story Points/Velocity

$52/20=2.6 \approx 3$  Sprints

Since the team has **3 months (6 Sprints)**, they can complete the app within time.

### Estimating Sprint Duration

Each Sprint is **2 weeks (14 days)**.

Total Duration Required=Number of Sprints × Sprint Duration

$=3 \times 14=42$  days

This means the team will **finish development within 42 days**, and the remaining time can be used for **testing and bug fixes**.

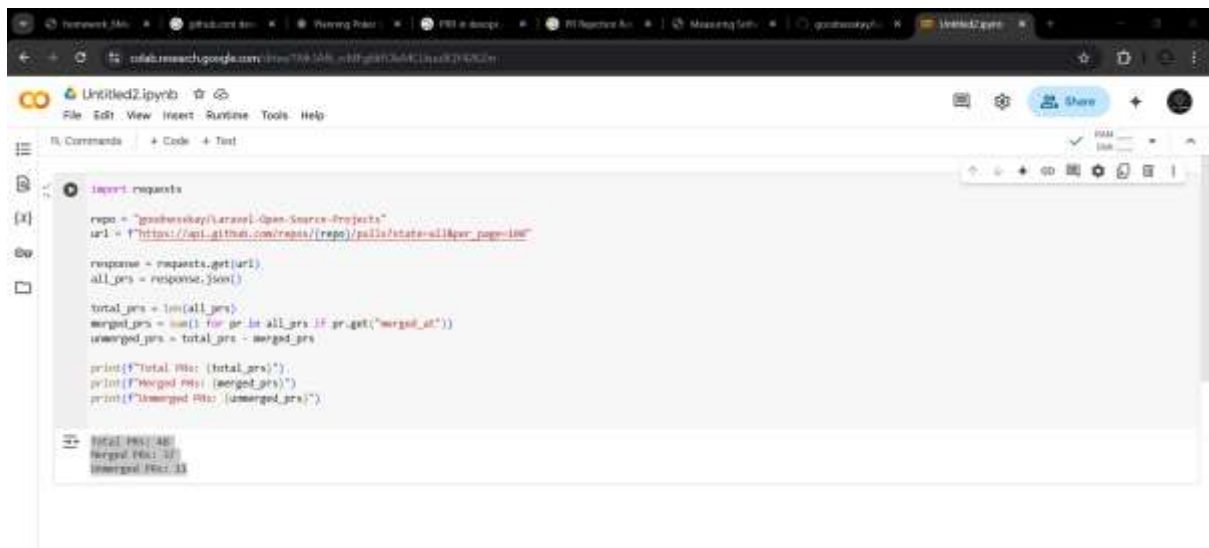
## Q2. A)

**PR Rejection Rate (PRR)** for the repository **Laravel-Open-Source-Projects** using the formula:

$$\text{PRR} = \text{Unmerged PRs} / \text{Merged PRs}$$

This will help determine the level of software development waste due to unused PRs. I'll fetch the necessary data and provide the results. Stay tuned! 😊

I have analyzed the Pull Request Rejection Rate (PRR) for the GitHub repository Laravel-Open-Source-Projects. This metric indicates the proportion of pull requests (PRs) that were not merged into the main project, highlighting potential inefficiencies or unused contributions.



The screenshot shows a Jupyter Notebook with a Python script that fetches pull request data from the GitHub repository 'gostheneskey/Laravel-Open-Source-Projects'. The script uses the GitHub API to get all pull requests, filters out merged ones, and calculates the total, merged, and unmerged counts. The output shows 48 total PRs, 37 merged PRs, and 11 unmerged PRs.

```
import requests

repo = "gostheneskey/Laravel-Open-Source-Projects"
url = f"https://api.github.com/repos/{repo}/pulls?state=all&per_page=100"

response = requests.get(url)
all_prs = response.json()

total_prs = len(all_prs)
merged_prs = sum(1 for pr in all_prs if pr.get("merged_at"))
unmerged_prs = total_prs - merged_prs

print(f"Total PRs: {total_prs}")
print(f"Merged PRs: {merged_prs}")
print(f"Unmerged PRs: {unmerged_prs}")
```

Output:

```
Total PRs: 48
Merged PRs: 37
Unmerged PRs: 11
```

### Data Overview:

- **Total PRs:** 48
- **Merged PRs:** 37
- **Unmerged PRs:** 11

### Calculation:

Using the formula:

$$\text{PRR} = \text{Unmerged PRs} / \text{Merged PRs}$$
$$= 11 / 37 = 0.29 \approx 29.7\%$$

This results in a PRR of approximately **30 %**.