

**SEB-312 Mobile Application Development**

**LAB # 11**

**LAB Title**

|  |
| --- |
| Implementing pagination to fetch data in batches. Adding infinite scrolling to a list |

**Assessment of CLO: 04, PLO: 05**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name:** |  | | |
| **Roll No.** |  | | |
| **Semester** |  | **Session** |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S. No.** | **Perf. Level**  **Criteria** | **Excellent**  **(2.5)** | **Good**  **(2)** | **Satisfactory**  **(1.5)** | **Needs Improvement**  **(0 ~ 1)** | **Marks Obtained** |
| **1** | Project Execution & Implementation | Fully functional, optimized, and well-structured. | Minor errors, mostly functional. | Some errors, requires guidance. | Major errors, non-functional, or not Performed. |  |
| **2** | Results & Debugging  Or Troubleshooting | Accurate results with effective debugging  Or Troubleshooting. | Mostly correct, some debugging Or Troubleshooting needed. | Partial results, minimal debugging  Or Troubleshooting. | Incorrect results, no debugging Or Troubleshooting, or not attempted. |  |
| **3** | Problem-Solving & Adaptability  (VIVA) | Creative approach, efficiently solves challenges. | Adapts well, minor struggles. | Some adaptability, needs guidance. | Lacks innovation or no innovation, unable to solve problems. |  |
| **4** | Report Quality & Documentation | Clear, structured, with detailed visuals. | Mostly clear, minor gaps. | Some clarity issues, missing details. | Poorly structured, lacks clarity, or not submitted. |  |
| **Total Marks Obtained Out of 10** | | | | | |  |

**Experiment evaluated by**

|  |  |  |  |
| --- | --- | --- | --- |
| **Instructor’s Name** | **Sidra Khatoon** | | |
| **Date** |  | **Signature** |  |

**Objective**

The objective of lab is implementing pagination to fetch data in batches. Adding infinite scrolling to a list.

**Assessment:**

1. What is pagination in the context of a REST API?

Pagination is a technique used to split large datasets into smaller chunks (pages) so the client doesn't fetch all data at once.  
Instead of returning all records, the API sends a specific number per request, often controlled by query parameters like page and limit.

2. How do you implement pagination in a Flutter app with a REST API?

You implement pagination in Flutter by:

Making an HTTP request with page and limit.

Using a ListView with a ScrollController to detect when the user reaches the bottom.

Calling the API again to load the next page and appending new data to the existing list.

3. What is the role of query parameters like page and limit in pagination?

These query parameters tell the server which chunk of data to return:

page=2 tells the API to return the second set of data.

limit=10 tells the API to return 10 items per page.

Example:

http

Copy code

GET /posts?page=2&limit=10

4. How do you handle pagination data in a REST API response in Flutter?

In Flutter:

Store the current page, limit, and a list of fetched items.

After fetching new data, check if the response has fewer items than limit → it means it's the last page.

Append new data to the existing list using setState() or a state management solution like Provider or Bloc.

5. How do you display paginated data in a Flutter ListView?

Use ListView.builder() with a ScrollController:

Load more data when the user scrolls near the bottom.

Add a loader (CircularProgressIndicator) at the end of the list while new data is being fetched.

Append the new page of data to the existing list.