1️⃣ Create a Slack App

1. Go to **Slack API: Your Apps**.
2. Click **Create New App** → **From scratch**.
3. Give it a name (e.g., MyBot) and choose the workspace you want to install it in.

2️⃣ Add Bot Functionality

1. In your app's settings, go to **Basic Information** → Scroll to **Add features and functionality**.
2. Click **Bots** and enable a bot user.
3. Give it a **display name** and **default username**.

3️⃣ Set Permissions (OAuth Scopes)

1. Go to **OAuth & Permissions** in your app settings.
2. Under **Bot Token Scopes**, add the scopes your bot needs, for example:
    - chat:write → send messages.
    - commands → respond to slash commands.
    - app_mentions:read → get notified when mentioned.
    - channels:history → read messages in public channels.
3. If you want slash commands:
    - Go to **Slash Commands** in the left menu.
    - Create a new command (e.g., /mybot) and set its **Request URL** to your backend endpoint.

4️⃣ Install the App to Your Workspace

1. Go to **OAuth & Permissions**.
2. Click **Install App to Workspace**.
3. Approve permissions.
4. You'll get a **Bot User OAuth Token** (starts with xoxb-...). Save it — you'll use it in your code.

5️⃣ Set up Event Subscriptions (optional, for real-time events)

1. Enable **Event Subscriptions**.
2. Enter your public **Request URL** (must be HTTPS).
3. Subscribe to events like:
    - message.channels (public channel messages)
    - app_mention (when someone mentions your bot)

6️⃣ Write Your Bot Code

- You can use **Python + FastAPI**, **Flask**, or **Bolt for Python** (Slack's official SDK).

7️⃣ Deploy Your Bot

- Host on a platform with HTTPS:
    - Render
    - Vercel
    - AWS Lambda
    - Railway
- Update Slack's settings with your live endpoint URL.

8️⃣ Test It

- Type /mybot or mention your bot in Slack.
- Check that it responds as expected.

Overview

This document outlines the functionality, user flow, and technical requirements for implementing three Slack slash commands:

- /ask
- /get_recommended_budget
- /schedule_report

The bot will use **Slack Slash Commands**, **modals**, and **external API integrations**.

1. /ask Command

Purpose

The user should be able to get answers form the knowledge agent.

User Flow

User types "***/ask"*** in slack

FE call the BE API

Slack opens up a modal prompting the user to enter the question

User enter the question and click submit

The question in passed to the BE

The BE fetches the result form the knowledge agent and pass it to FE as text

FE display the result in the message filed as slash command response

Example:
Question : What is lifesight?
Answer: Lifesight is a unified marketing measurement platform designed to help modern marketers make better decisions using modern methodologies that go beyond conventional touch-based attribution.

2./create_planner command

Purpose

To create a planner based on the parameters entered by user

User Flow

User types "**/create_planner"** in slack

FE call the BE API

Slack opens up a modal with all necessary fields for input

User enter all the details and click submit

All inputs are passed to the BE

The BE creates the planner according to the user inputs and pass the message " Planner created successfully"

FE display the result in the message filed as slash command response

```
POST /mmm/plan/recommended-budget
Body: {
  "modelId": "model1",
  "solId": "5_645_55",
  "timePeriod": "1 month",
  "optimisedPeriod": "1 month",
  "constraintType": "conservative"
}

Message:  Planner created successfully
```

3./Schedule_reports

Purpose

Alows users to schedule automated reports export

User Flow

User types "***/schedule_reports"*** in slack

FE call the BE API

Slack opens up a modal with all necessary fields for input

User enter all the details and click submit

All inputs are passed to the BE

The API schedules the report export according to the user's requirements

FE display the message "Report Schedule Successful" in the message field

```
POST /api/v1/dashboard/exports
Body: {
  "report": "...",
  "dashboard": "...",
  "frequency": "..."
}

Message: Report Scheduled Successfully
```

Requirments

Bot Token Scopes:

- **commands**

- **chat:write**
- **users:read**

Slash Commands:

- **/ask → POST to /slack/ask**
- **/get_recommended_budget → POST to /slack/get-recommended-budget**
- **/schedule_report → POST to /slack/schedule-report**

Slack Command	Bot Endpoint	Function
/ask	**/slack/ask**	**Sends question to Ask API and returns answer**
/get_recommended_budget	**/slack/get-recommended-budget**	**Opens modal, processes submission, calls Recommended Budget API**
/schedule_report	**/slack/schedule-report**	**Opens modal, processes submission, calls Report Scheduling API**

- ❖ On hitting the slash command the BE should respond within 3 sec or immediately send a short 200 OK response and then use the **response**

- ❖ The API's return should be in specific format
- ❖ The BE should design each modal according to the API needs
- ❖ Slack **Interactivity & Shortcuts** should be active
- ❖
- ❖