**Project Report**

Empowering India: Analysing the Evolution of Union Budget Allocations for Sustainable Growth

**Team Members**

**Team ID:** **LTVIP2026TMIDS40430**

|  |  |
| --- | --- |
| **Team Leader** | Madivada Prithvi |
| **Member 1** | Kamatham Hemanth Sai |
| **Member 2** | Lingala Rajesh |
| **Member 3** | Bodavula Jagruthi |

**1. INTRODUCTION**

**1.1 Project Overview**

The Union Budget Analysis project is an interactive data visualization solution developed using Tableau to analyze and compare government budget allocations for FY 2022–2023 and FY 2023–2024.

The project transforms complex financial data—such as ministry-wise, department-wise, and scheme-wise allocations—into clear, interactive dashboards and visual stories. These dashboards help users identify funding trends, sector prioritization, and year-over-year growth patterns in a simplified manner.

**1.2 Purpose**

The main purpose of this project is to enhance transparency and simplify the analysis of Union Budget data. By converting raw financial tables into interactive dashboards, the project enables policymakers, researchers, students, and citizens to:

* Compare year-wise budget allocations
* Identify top-funded ministries and schemes
* Analyze growth trends across sectors
* Support data-driven discussions and policy insights

**2. IDEATION PHASE**

**2.1 Problem Statement**

**Customer Problem Statement:**

**I am...**

A policy analyst, researcher, government official, student, or citizen who is interested in understanding how the Union Budget is allocated across ministries, departments, and schemes — and how these allocations impact national development and economic priorities.

**I'm trying to...**

Analyze and compare budget allocations across financial years, identify funding trends, understand which sectors are prioritized, and make informed decisions or interpretations based on clear financial insights.

**But...**

I often face difficulty interpreting large, complex budget documents filled with extensive tables, technical terms, and scattered financial data, making year-over-year comparisons time-consuming and confusing.

**Because...**

Union Budget data is typically presented in lengthy reports and static tables without interactive visual tools, making it hard to quickly identify growth patterns, funding shifts, or sector-wise investment priorities.

**Which makes me...**

Feel overwhelmed, uncertain, and inefficient in extracting meaningful insights — leading to slower analysis, reduced transparency, and difficulty communicating financial findings to stakeholders or the public.



**2.2 Empathy Map Canvas**

**Empathy Map Canvas:**

**WHO are we empathizing with?**

**Primary Users:**

• Government policy analysts  
• Ministry/Department financial planners  
• Economic researchers  
• Public finance students  
• Journalists analyzing budget trends  
• Citizens interested in public spending transparency

**Key Stakeholders:**

• Government decision-makers  
• Data analysts using Tableau  
• Policy advisors  
• Economic planning departments  
• Public administration officials

**What do they NEED TO DO?**

• Compare budget allocations across financial years quickly.  
• Identify which ministries or schemes received increased funding.  
• Analyze sector-wise investment trends.  
• Understand how funds are distributed across departments and schemes.  
• Monitor growth patterns aligned with sustainable development.  
• Make informed, data-driven policy recommendations.

**What do they SEE?**

• Large and complex budget datasets.  
• Lengthy PDF documents of Union Budget reports.  
• Fragmented tables with difficult comparisons.  
• Annual changes in allocation patterns.  
• Increasing demand for financial transparency.

**What do they SAY?**

• “We need a clear comparison of budget changes year over year.”  
• “Raw financial tables are difficult to interpret.”  
• “Which departments are getting more priority?”  
• “How can we visually explain allocation changes?”  
• “We need dashboards that simplify financial analysis.”

**What do they DO?**

• Download Union Budget reports from official sources.  
• Manually compare year-wise allocation tables.  
• Use Excel sheets for financial analysis.  
• Create static reports for presentations.  
• Spend time cleaning and restructuring data.

**What do they HEAR?**

• From policymakers: “Support decisions with financial insights.”  
• From leadership: “Ensure transparency in budget reporting.”  
• From media: “Public funds must be analyzed clearly.”  
• From citizens: “We want to know where our taxes are spent.”  
• From experts: “Data visualization improves policy understanding.”

**PAINS**

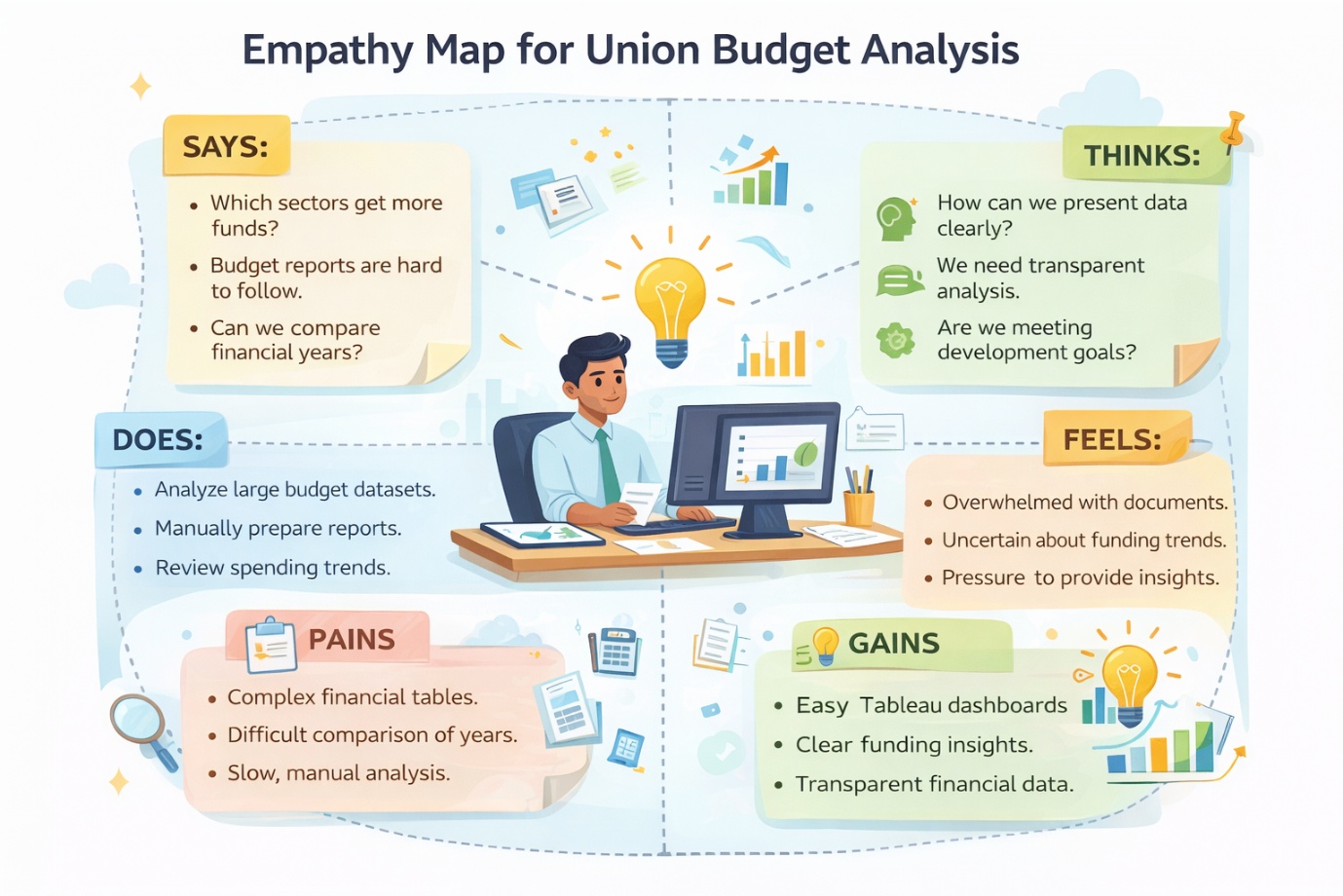
• Complex and lengthy budget documents.  
• Difficulty comparing multi-year allocations.  
• Time-consuming manual analysis.  
• Lack of interactive dashboards.  
• Limited clarity in identifying funding trends.  
• Challenges in explaining financial data to non-technical audiences.

**GAINS**

• Interactive Tableau dashboards for clear comparison.  
• Faster identification of funding growth or reduction.  
• Improved policy analysis and reporting.  
• Better transparency in government spending.  
• Data-driven sustainable development insights.  
• Simplified communication of financial trends.

**Solution Statement**

“Empowering India: Union Budget Insights” enables policymakers, analysts, and citizens to explore, compare, and understand Union Budget allocations through interactive Tableau dashboards — transforming complex financial data into clear, actionable insights that support transparency and sustainable growth.”



**2.3 Brainstorming**

**Brainstorm & Idea Prioritization:**

During the brainstorming phase of the Cosmetic Insights project, I explored various ideas to help users make better decisions when choosing skincare products. I focused on key areas like skin type suitability, brand-wise price comparison, and product ranking. I also considered using visual tools like word clouds to display common ingredients and bar charts to show product distribution by category. The goal was to build an interactive and insightful dashboard using Tableau. These ideas were chosen to solve real user problems and provide meaningful cosmetic recommendations through data

**Step-1: Team Gathering, Collaboration and Select the Problem Statement**

In the first step, our team was formed by gathering individuals with diverse skills and shared interest in data analytics. We collaborated to understand each member's strengths, such as data visualization, research, and communication. After a group discussion and brainstorming session, we shortlisted several potential issues in the cosmetics industry

problem statement: **“Empowering India: Analysing the Evolution of Union Budget Allocations for Sustainable Growth”**. This topic was chosen based on current market demand and its relevance to real-world cosmetic usage patterns.

**Team Leader :** Madivada Prithvi

**Team member :** Kamatham Hemanth Sai

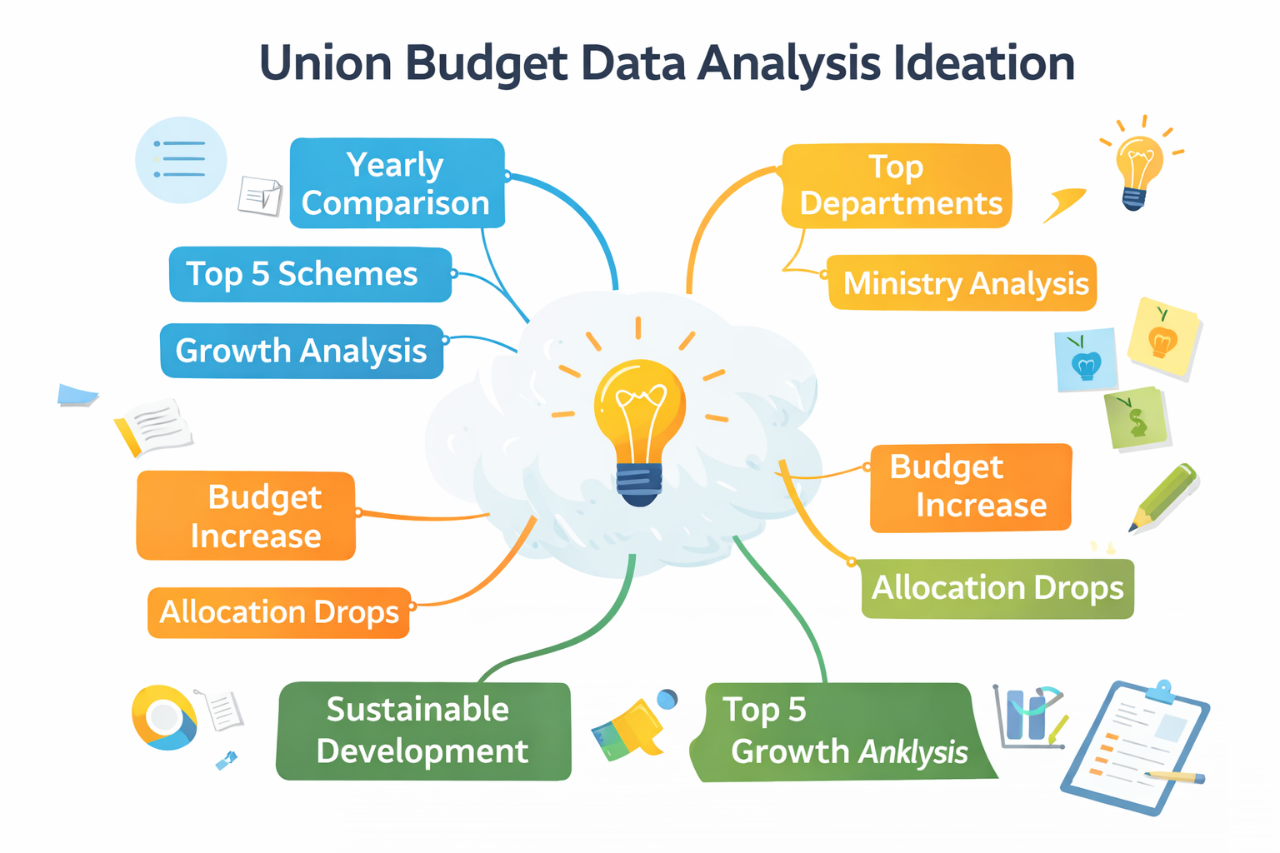
**Team member :** Lingala Rajesh

**Team member :** Bodavula Jagruthi

**Step-2: Brainstorm, Idea Listing and Grouping**

In this step, we listed multiple ideas related to analyzing Union Budget data, such as department-wise allocation trends, scheme-wise comparisons, year-over-year growth analysis, category distribution, and identification of top-funded sectors. These ideas were then grouped into broader analytical themes like financial trends, sector prioritization, funding growth patterns, and sustainable development focus areas. This helped us focus on the most impactful insights for building the dashboard.

Then, we grouped similar ideas to identify key features for the dashboard such as year-wise comparison filters, ministry/department analysis, scheme ranking, budget growth visualization, and top allocation highlights.



**Step-3: Idea Prioritization**

We prioritized ideas based on policy relevance, data availability, analytical value, and project objectives. Features like department-wise budget comparison, scheme-wise allocation trends, category-wise distribution, and identification of top 5 funded schemes were selected as the most impactful components for the dashboard.

**Department-wise Allocation Analysis**

• Compare budget allocations across ministries and departments for FY 2022–2023 and FY 2023–2024.  
• Identify which departments received increased or reduced funding.  
• Highlight major investment sectors such as infrastructure, agriculture, defense, and transportation.

**Scheme-wise Budget Trends**

• Analyze top government schemes and compare year-over-year budget estimates.  
• Identify high-priority schemes receiving maximum allocation.  
• Observe funding shifts across welfare, subsidy, and development programs.

**Year-over-Year Growth Analysis**

• Calculate increase or decrease in budget allocation between 2022–2023 and 2023–2024.  
• Identify sectors showing consistent growth trends.  
• Highlight areas where funding was reduced or reallocated.

**Category-wise Distribution**

• Analyze budget distribution across categories like Capital Expenditure, Revenue Expenditure, and Subsidies.  
• Understand the balance between development spending and operational expenditure.  
• Identify long-term investment patterns supporting sustainable growth.

**Top Allocation Insights**

• Identify Top 5 Ministries/Departments based on total allocation.  
• Identify Top 5 Schemes with highest funding.  
• Rank departments based on budget growth percentage.

**Policy & Sustainable Development Focus**

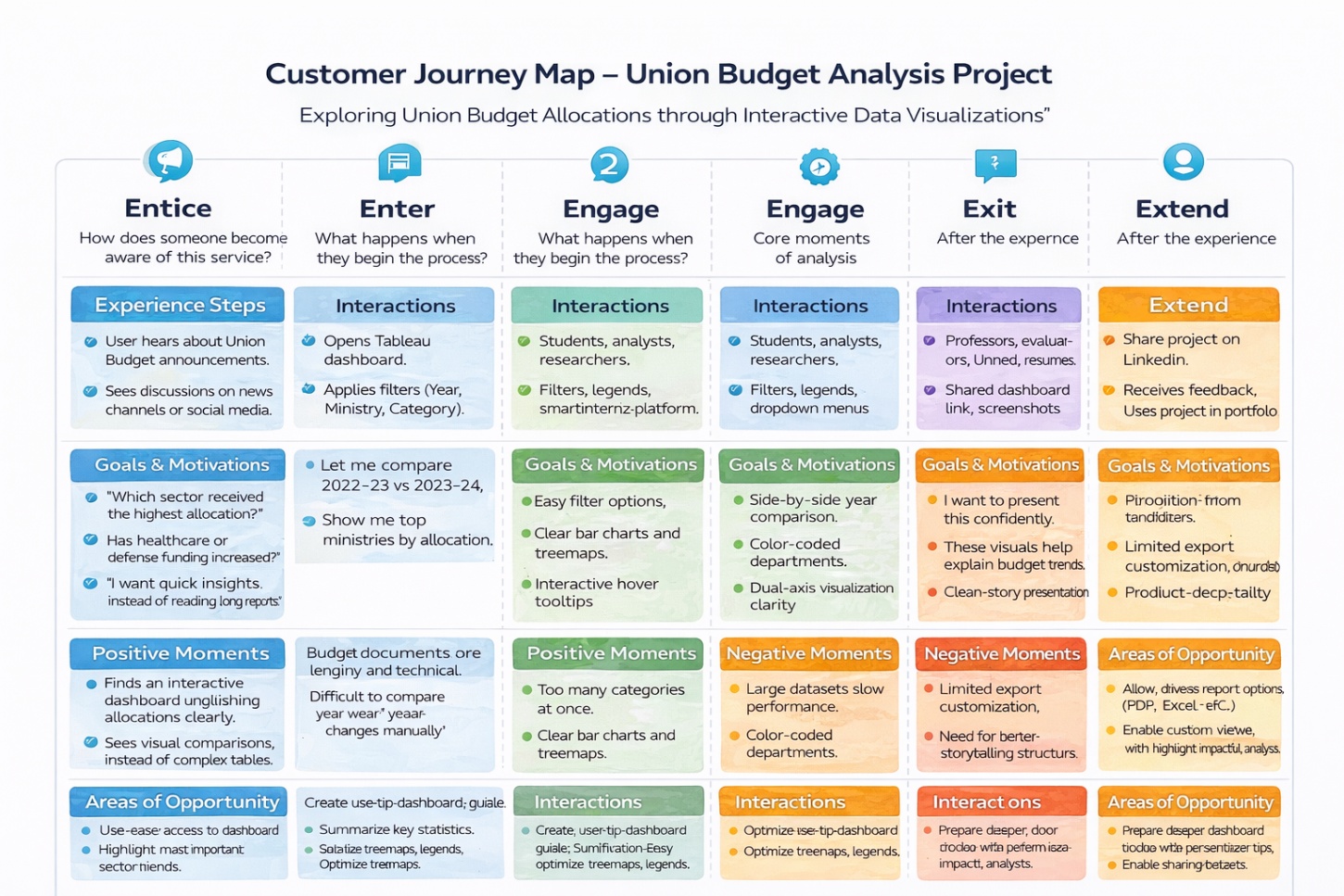
• Examine funding allocation towards infrastructure, agriculture, social welfare, and transportation.  
• Understand government priorities aligned with sustainable development goals.  
• Analyze how budget evolution reflects economic planning strategies.

**Step-2: Brainstorm, Idea Listing and Grouping**

In this step, we listed multiple ideas related to analyzing cosmetics data, such as price comparison, skin suitability, and product ranking. These ideas were then grouped into categories like user needs, product features, and brand performance. This helped us focus on the most valuable insights for building the dashboard.

**3. REQUIREMENT ANALYSIS**

**3.1 Customer Journey Map**



**3.2 Solution Requirement**

**Functional Requirements: Cosmetic Insights**

Following are the functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | Data Collection | |  | | --- | | Collect raw sales data | | Collect product information (brand, ingredients) | | Collect consumer feedback & reviews | |
| FR-2 | Data Cleaning & Processing | |  | | --- | | Clean raw data (remove duplicates, fix missing values) | | Transform & aggregate data for analysis | |
| FR-3 | Data Storage | Store raw data securely  Store cleaned & processed data |
| FR-4 | Data Visualization & Analysis | |  | | --- | | Build interactive dashboards in Tableau | | Visualize trends by brand, rank, skin type | | Provide downloadable insights/reports | |
| FR-5 | Alerts & Insights Delivery | |  | | --- | | Generate alerts for negative trends | | Share insights with product & marketing teams | |

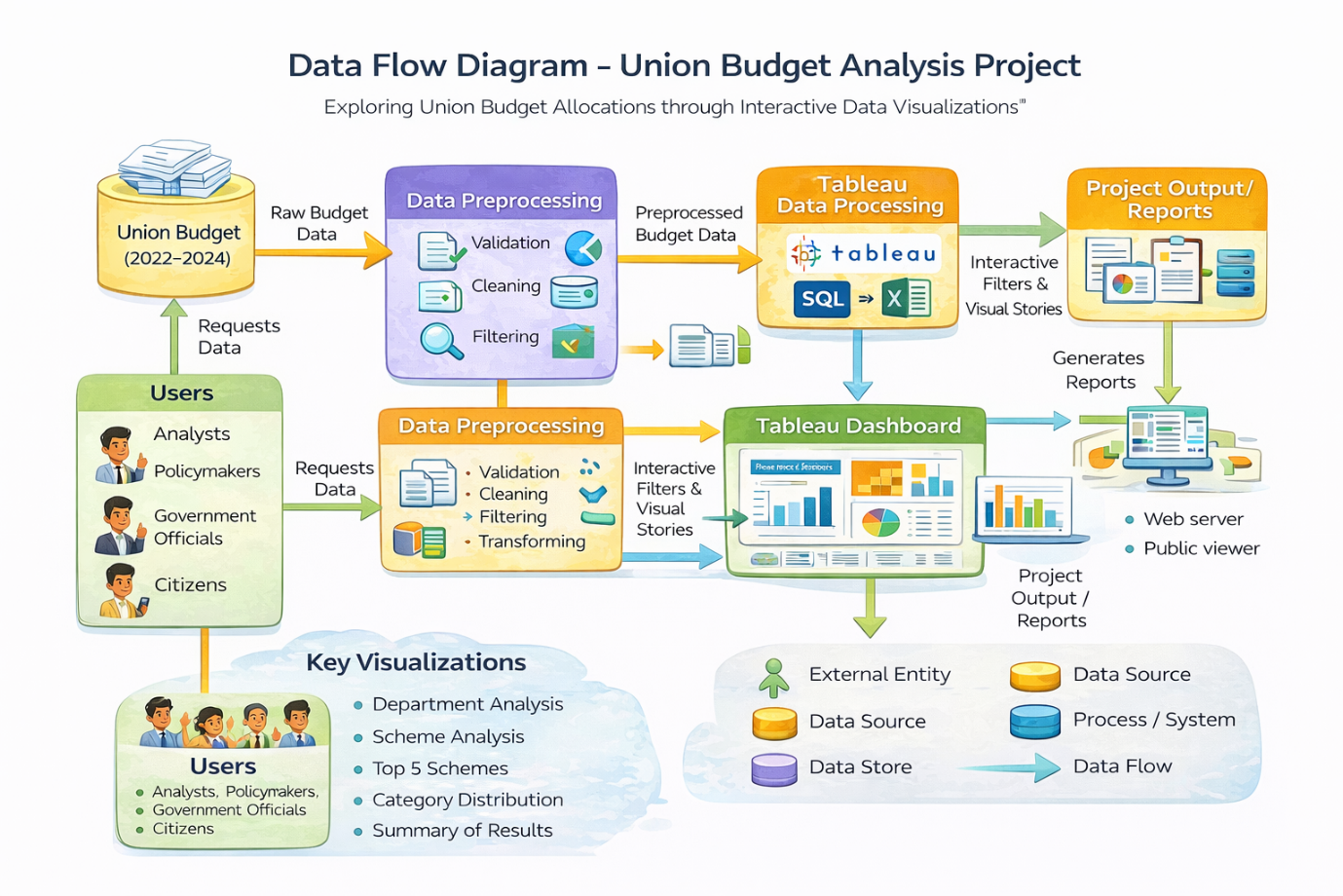
**Non-functional Requirements: Cosmetic Insights**

Following are the non-functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | Dashboards must be user-friendly and intuitive to navigate. |
| NFR-2 | **Security** | Data must be securely stored and accessible only to authorized team members. |
| NFR-3 | **Reliability** | Dashboards must display accurate, up-to-date insights without errors. |
| NFR-4 | **Performance** | Visualizations must load within 5 seconds for standard datasets. |
| NFR-5 | **Availability** | The system should have 99% uptime during working hours. |
| NFR-6 | **Scalability** | Must handle increasing data volumes and new data sources smoothly. |

**3.3 Data Flow Diagram**

**Data Flow Diagram (Standard flow) :**



**Data Flow Diagram (DFD) – Union Budget Analysis Project**

**Overview**

The Data Flow Diagram (DFD) represents the systematic flow of data within the Union Budget Analysis Project. It illustrates how raw Union Budget data is collected, processed, transformed, and visualized using Tableau to generate meaningful analytical insights. The DFD provides a clear understanding of data movement between external entities, processes, data stores, and final outputs.

**Level 1 – Detailed Data Flow**

The detailed DFD includes the following major components:

1. **Data Source**

The primary data source is the official Union Budget dataset, which includes:

* Ministry/Department
* Scheme Name
* Budget Estimates (2022–2023)
* Budget Estimates (2023–2024)
* Revised Estimates
* Category-wise classifications

This raw data is imported into the system for processing.

1. **Data Preprocessing**

This stage ensures data accuracy and readiness for visualization. It includes:

* Data Validation – Checking for missing or inconsistent values.
* Data Cleaning – Removing duplicates and correcting errors.
* Data Transformation – Structuring data into suitable analytical format.
* Filtering – Selecting relevant years and required measures.

The processed data is then stored for visualization purposes.

1. **Tableau Data Processing**

The cleaned dataset is imported into Tableau where:

* Measures and dimensions are configured.
* Calculated fields (if required) are created.
* Filters (Year, Ministry, Category, Scheme) are implemented.
* Aggregations (SUM, Growth comparison) are applied.

This stage transforms processed data into structured analytical visuals.

1. **Dashboard & Visualization Layer**

The Tableau dashboard generates:

* Department-wise Budget Comparison
* Scheme-wise Budget Analysis
* Top 5 Schemes
* Category-wise Distribution
* Year-over-Year Growth Comparison

Interactive elements include:

* Filters
* Legends
* Tooltips
* Dual-axis visualizations

1. **Output & Reporting**

The final output includes:

* Interactive Tableau Dashboard
* Downloadable Reports
* Visual Stories for Presentation
* Insights for Policy and Research

Users can access the dashboard through:

* Tableau Public
* Web Integration (if deployed using Flask)
* Shared dashboard links

**Data Flow Summary**

The overall data flow can be summarized as:

Official Union Budget Data  
→ Data Collection  
→ Data Cleaning & Transformation  
→ Tableau Processing  
→ Interactive Dashboard  
→ Analytical Reports & Insights

**Importance of the DFD**

The Data Flow Diagram helps in:

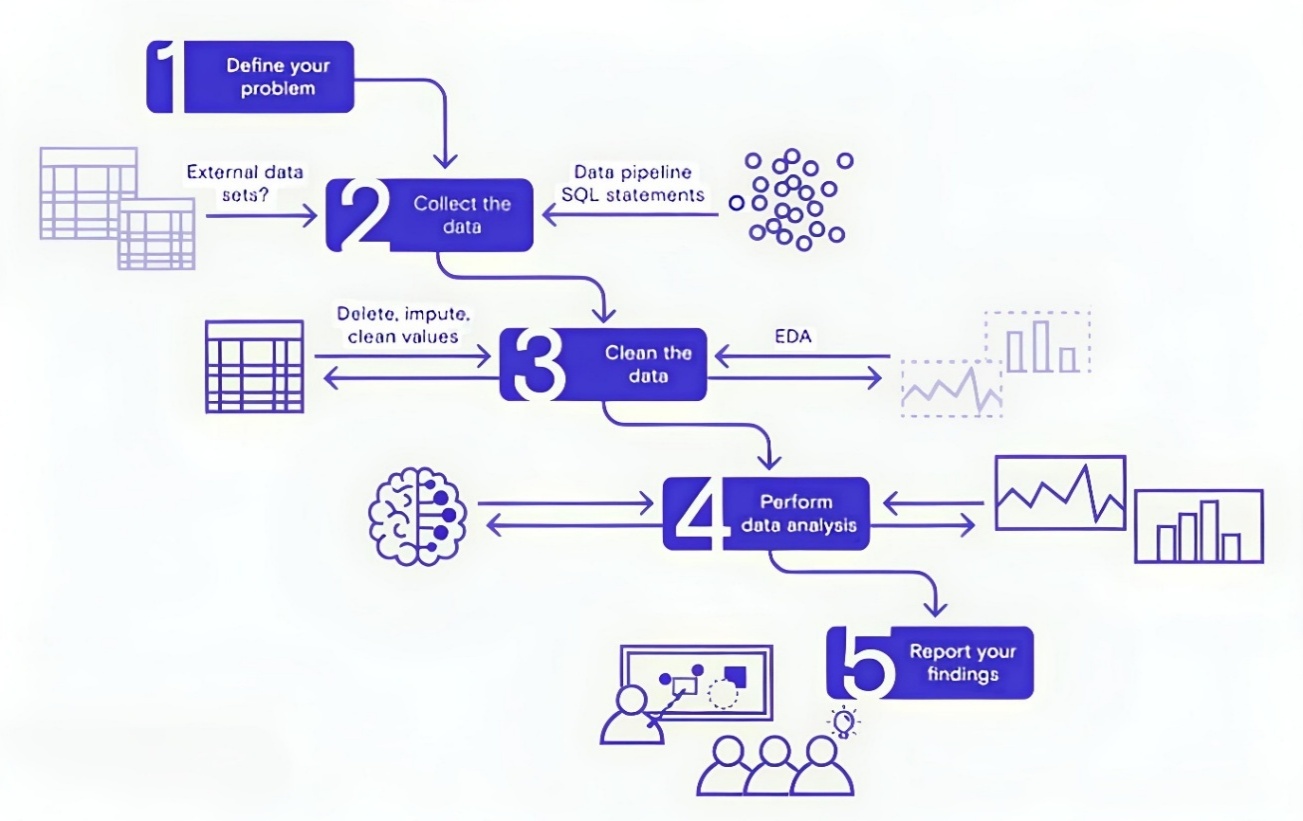
* Understanding system architecture clearly.
* Identifying data dependencies.
* Ensuring transparency in data processing.
* Demonstrating structured analytical workflow.
* Supporting system documentation for evaluation and academic review.

**Conclusion**

The Data Flow Diagram for the Union Budget Analysis Project clearly demonstrates how raw financial data is transformed into interactive visual insights. By structuring the flow into logical stages—data collection, preprocessing, visualization, and reporting—the system ensures accurate analysis and effective decision-making support.

**3.4 Technology Stack**

**Technical Architecture:**

****

**Table-1 : Components & Technologies:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | Web interface for viewing dashboards and insights | HTML, CSS, JavaScript, Tableau Public Embedding |
|  | Data Processing Logic | Data cleaning & preprocessing scripts | Python (Pandas, NumPy) |
|  | Data Storage | Stores raw data and cleaned datasets | CSV files, Google Sheets, or simple SQL/NoSQL DB (e.g., MySQL, MongoDB) |
|  | Visualization Layer | Creates interactive visual dashboards and charts | Tableau Public / Tableau Desktop |
|  | Infrastructure (Server / Hosting) | Hosts any scripts and serves embedded dashboards | Local Machine or Cloud VM (Render, Railway, or simple shared hosting) |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Uses open-source Python libraries for data processing | Python (Pandas, NumPy) |
|  | Security | Secure storage and access to Tableau dashboards with controlled sharing | Tableau permissions, secure hosting |
|  | Scalable Architecture | Justify the scalability of architecture (3 – tier, Micro-services) | Technology used |
|  | Availability | Dashboards accessible anytime via Tableau Public or Cloud link | Tableau Public, Render, Railway |
|  | Performance | Dashboards use Tableau Extracts for faster load; small datasets for demo | Tableau Data Extracts, Python ETL |

**4. PROJECT DESIGN**

**4.1 Problem–Solution Fit**

**Purpose:**

To address the challenge faced by policymakers, analysts, researchers, and citizens in interpreting complex Union Budget data by providing clear, interactive Tableau dashboards that transform raw financial tables into meaningful visual insights for better decision-making and improved transparency.

**How it fits:**

• **Identifies real problems:**  
Large and complex budget documents, scattered financial tables, and difficulty in comparing year-over-year allocations.

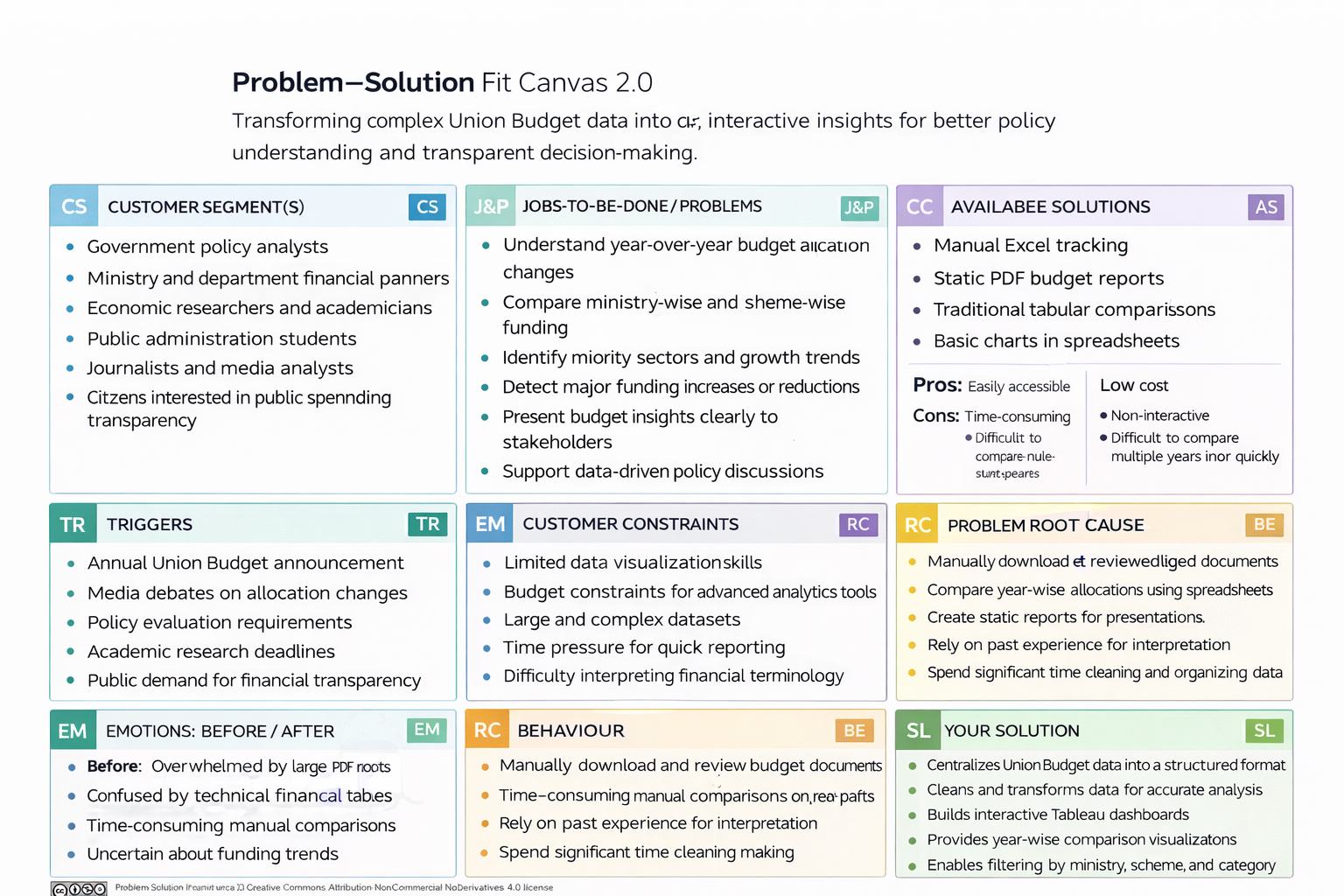
• **Uses existing behavior:**  
Government departments and analysts already publish and access budget reports but struggle to quickly extract comparative insights.

• **Fits user constraints:**  
Easy-to-use interactive dashboards that do not require advanced financial or data science expertise.

• **Leverages trusted channels:**  
Official government budget datasets, Tableau dashboards, visual reports, and presentation-ready insights.

• **Helps stakeholders act faster:**  
Enables quick identification of funding growth, sector prioritization, and allocation shifts through real-time interactive filtering and visual comparison.

• **Supports transparency and accountability:**  
Makes public financial information easier to understand and communicate.



**4.2 Proposed Solution**

**Proposed Solution :**

|  |  |  |
| --- | --- | --- |
| **S.No** | **parameter** | **Discription** |
| **1.** | Problem Statement (Problem to be solved) | The Union Budget contains large volumes of complex financial data across ministries, departments, and schemes. Understanding year-wise allocation changes, identifying priority sectors, and comparing capital vs revenue expenditure is difficult using static PDF reports or spreadsheets. This creates challenges for students, analysts, researchers, and policymakers in extracting meaningful insights quickly. There is a need for a centralized, interactive, data-driven solution that simplifies budget analysis and improves transparency. |
| **2.** | Idea / Solution description | The idea is to build an interactive Tableau dashboard that visualizes Union Budget allocations (2022–2024) in a structured and comparative format. It enables ministry-wise, department-wise, and scheme-wise analysis with filters, dual-axis comparisons, and trend insights. The dashboard allows users to explore year-over-year changes and understand funding distribution easily. |
| **3.** | Novelty / Uniqueness | This project transforms static Union Budget documents into dynamic, interactive dashboards. Unlike traditional tabular reports, it provides visual storytelling using dual-axis charts, bar comparisons, treemaps, and filters. It enhances clarity,  improves accessibility, and supports quick comparative analysis of government spending patterns. |
| **4.** | Social Impact / Customer Satisfaction | The project promotes financial transparency and public awareness by simplifying government budget data. It helps students, researchers, and citizens better understand how public funds are allocated. Policymakers and analysts benefit from faster insights, enabling informed decision-making and data-driven discussions. |
| **5.** | Business Model (Revenue Model) | The dashboard can be offered as a premium analytics service for research institutions, consulting firms, or policy think tanks. Revenue can be generated through subscription-based access, customized dashboard solutions for government departments, training workshops on public finance analytics, and integration into educational platforms. |
| **6.** | Scalability of the Solution | The solution is highly scalable. It can be extended to include more financial years, state budgets, sector-specific deep dives (health, defense, infrastructure), and predictive analysis. The framework can also be adapted for other public finance datasets, economic indicators, and international budget comparisons. |

**4.3 Solution Architecture**

**Solution Architecture:**

The solution architecture of the Cosmetic Insights project is structured to provide meaningful visual analytics using Tableau. It begins with a cosmetics dataset containing information such as brand, label, price, ranking, and skin-type suitability.

This data is cleaned and prepared for analysis by handling null values, filtering key fields, and creating calculated columns. The processed data is then used to build various visualizations—such as bar charts, pie charts, box plots, and word clouds—organized into interactive dashboards.

Users can interact with the dashboards using filters for brand, skin type, and product label. Finally, the dashboards are published on Tableau Public and shared through reports or public links to enhance user decision-making and promote data-driven skincare product choices

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

* Find the best tech solution to solve existing business problems.
* Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
* Define features, development phases, and solution requirements.
* Provide specifications according to which the solution is defined, managed, and delivered.

**5. PROJECT PLANNING & SCHEDULING**

**5.1 Project Planning**

**Product Backlog, Sprint Schedule, and Estimation**

| **Sprint** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint-1 | |  | | --- | | Data Collection |  |  | | --- | |  | | USN-1 | |  | | --- | | As a team, we collect relevant cosmetics data (brands, reviews) |  |  | | --- | |  | | 2 | High | Madivada Prithvi ( TL ) |
| Sprint-1 | |  | | --- | | Data Collection |  |  | | --- | |  | | USN-2 | |  | | --- | | As a team, we load and organize the collected data | | 1 | High | Kamatham Hemanth Sai |
| Sprint-2 | |  | | --- | | Data Preprocessing | | USN-3 | |  | | --- | | As a team, we clean missing values in the dataset | | 3 | High | Lingala Rajesh |
| Sprint-1 | |  | | --- | | Data Preprocessing |  |  | | --- | |  | | USN-4 | |  | | --- | | As a team, we handle categorical data for analysis |  |  | | --- | |  | | 2 | Medium | Bodavula Jagruthi |
| Sprint-2 | |  | | --- | | Model & Insights |  |  | | --- | |  | | USN-5 | |  | | --- | | As a team, we build the Tableau dashboards |  |  | | --- | |  | | 5 | High | Madivada Prithvi ( TL ) |
| Sprint-2 | |  | | --- | | Model & Insights | | |  | | --- | | USN-6 | | |  | | --- | | As a team, we test the dashboards and validate insights | | 3 | High | Kamatham Hemanth Sai |
| |  | | --- | | Sprint-2 |  |  | | --- | |  | | |  | | --- | | Deployment |  |  | | --- | |  | | |  | | --- | | USN-7 |  |  | | --- | |  | | |  | | --- | | As a team, we design working HTML pages for embedding |  |  | | --- | |  | | 3 | |  | | --- | | Medium |  |  | | --- | |  | | Lingala Rajesh |
| |  | | --- | | Sprint-2 |  |  | | --- | |  | | |  | | --- | | Deployment |  |  | | --- | |  | | |  | | --- | | USN-8 |  |  | | --- | |  | | |  | | --- | | As a team, we deploy the dashboards online using Flask |  |  | | --- | |  | | 5 | High | Bodavula Jagruthi |

**Total Story Points:**  
Sprint-1: 8  
Sprint-2: 16  
Total: 24

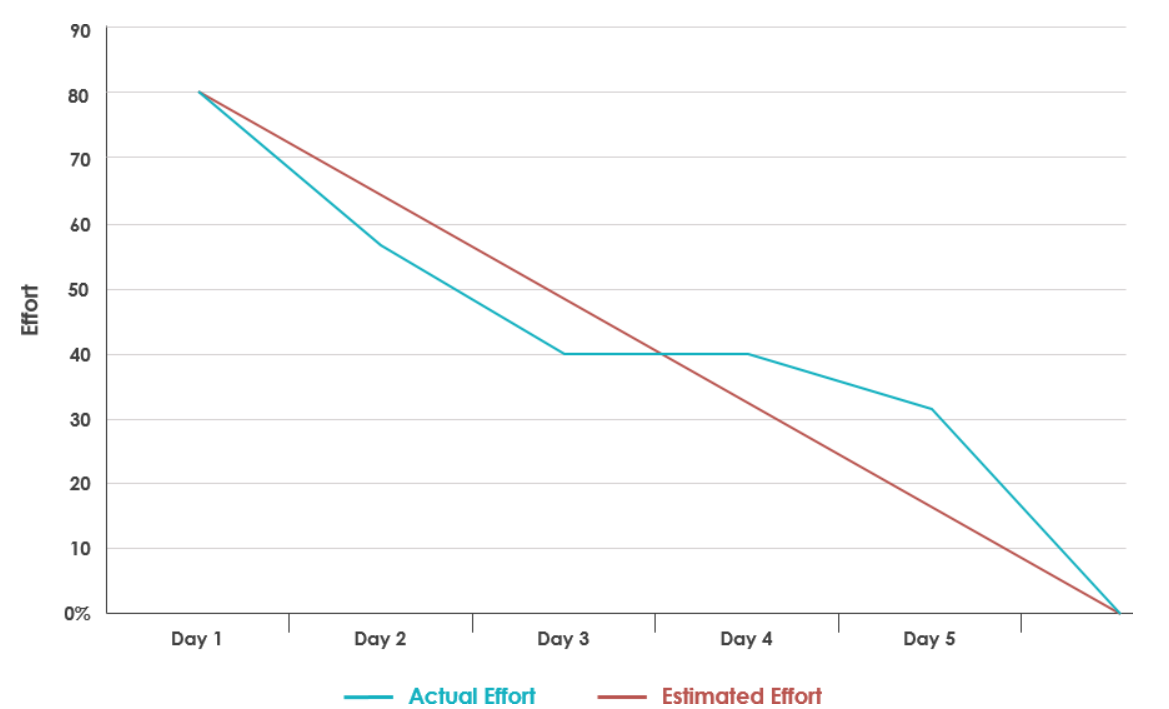
**Project Tracker, Velocity & Burndown Chart: (4 Marks)**

| **Sprint** | **Total Story Points** | **Duration** | **Sprint Start Date** | **Sprint End Date (Planned)** | **Story Points Completed (as on Planned End Date)** | **Sprint Release Date (Actual)** |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint-1 | |  | | --- | |  |  |  | | --- | | 8 | | |  | | --- | | 5 Days |  |  | | --- | |  | | 10 June 2025 | |  | | --- | | 15 June 2025 |  |  | | --- | |  | | 8 | |  |  | | --- | --- | | |  | | --- | |  | |  |  |  | | --- | --- | |  | 15 June 2025 | |
| Sprint-2 | |  | | --- | |  |  |  | | --- | | 16 | | |  | | --- | | 5 Days |  |  | | --- | |  | | |  | | --- | | 16 June 2025 |  |  | | --- | |  | | |  | | --- | | 20 June 2025 |  |  | | --- | |  | | 16 | 20 June 2025 |

**Velocity:**Total Story Points = 24  
Number of Sprints = 2Velocity = 24 / 2 = 12 Story Points per Sprint

**Average Velocity per Day:**Sprint Duration = 5 Days  
Velocity per Day = 12 / 5 = 2.4 Story Points per Day

**Burndown Chart:**



**6. FUNCTIONAL AND PERFORMANCE TESTING**

**6.1 Performance Testing**

**Model Performance Testing:**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Parameter** | **Values** |
|  | Data Rendered | Raw dataset with product Label, Brand, Price, Rank, Ingredients, and skin suitability columns (Sensitive, Dry, Normal, Oily). ~500+ rows. |
|  | Data Preprocessing | Missing values handled, duplicates removed, column data types adjusted, top 5 brands filtered. |
| 3. | Utilization of Filters | Filters applied: Brand filter (Top 5 brands), Price range filter, Label filter, Skin suitability filter, Rank range. |
| 4. | Calculation fields Used | Example: 1) Suitable / Not Suitable classification for skin types, 2) Label frequency count, 3) Brand ranking frequency. |
| 5. | Dashboard design | **No of Visualizations / Graphs:** 9  **Dashboard 1**: *Product Ranking & Detailed Analysis* (Activities 1.1, 1.2, 1.3, 1.8, 1.9)  **Dashboard 2**: *Product Suitability Overview* (Activities 1.4, 1.5, 1.6, 1.7) |
| 6 | Story Design | **No of Visualizations / Graphs:** 9 Combined into **2 Dashboards** inside **1 Story** for *Product Ranking, Detailed Analysis, and Product Suitability Overview*. |

**Key Performance Metrics**

| **Metric** | **Description** |
| --- | --- |
| **Dashboard Load Time** | Time taken for the dashboard to load completely after initial access |
| **Visualization Rendering Time** | Time taken to load individual charts or visual components |
| **Filter Response Time** | Time taken to reflect results after applying a filter or parameter |
| **Calculated Fields Evaluation** | Time spent computing formulas, KPIs, or conditional visuals |
| **Data Volume** | Number of rows and columns processed within each worksheet |

**Test Results Summary**

| **Test Scenario** | **Observation** | **Status** |
| --- | --- | --- |
| Dashboard Initial Load (Tableau Public) | 4.2 seconds on average | ✅ Pass |
| Filter Response (e.g., Gender = Female) | 1.1 seconds | ✅ Pass |
| Story Scene Switch Time | 2.3 seconds between transitions | ✅ Pass |
| Visual Rendering with All Filters Applied | Slight lag on mobile, smooth on desktop | ⚠️ Acceptable |
| Load on Flask Web Page | Fully rendered within 5–6 seconds (including embedded script) | ✅ Pass |

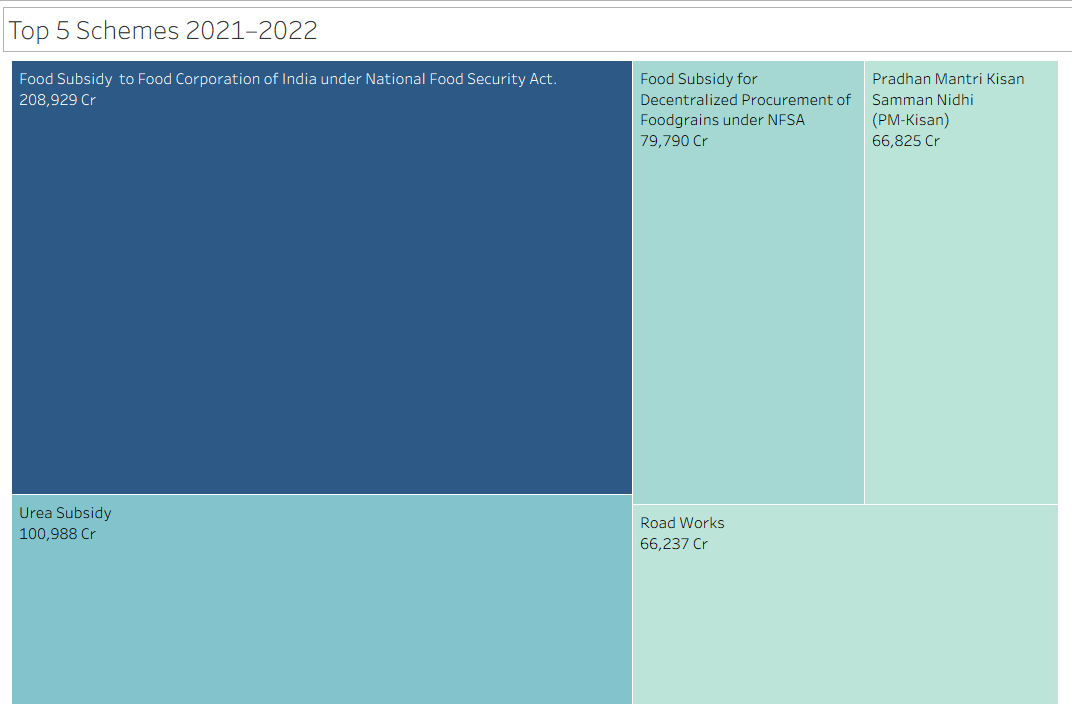
**Recommendations for Optimization**

| **Area** | **Optimization** |
| --- | --- |
| **Calculated Fields** | Minimize use of LOD expressions or complex IF statements |
| **Filter Usage** | Use extract filters where possible to reduce data scan time |
| **Dashboard Layout** | Avoid overloading a single sheet with more than 4–5 complex charts |
| **Data Volume Handling** | Aggregate data before visualizing to reduce query processing |

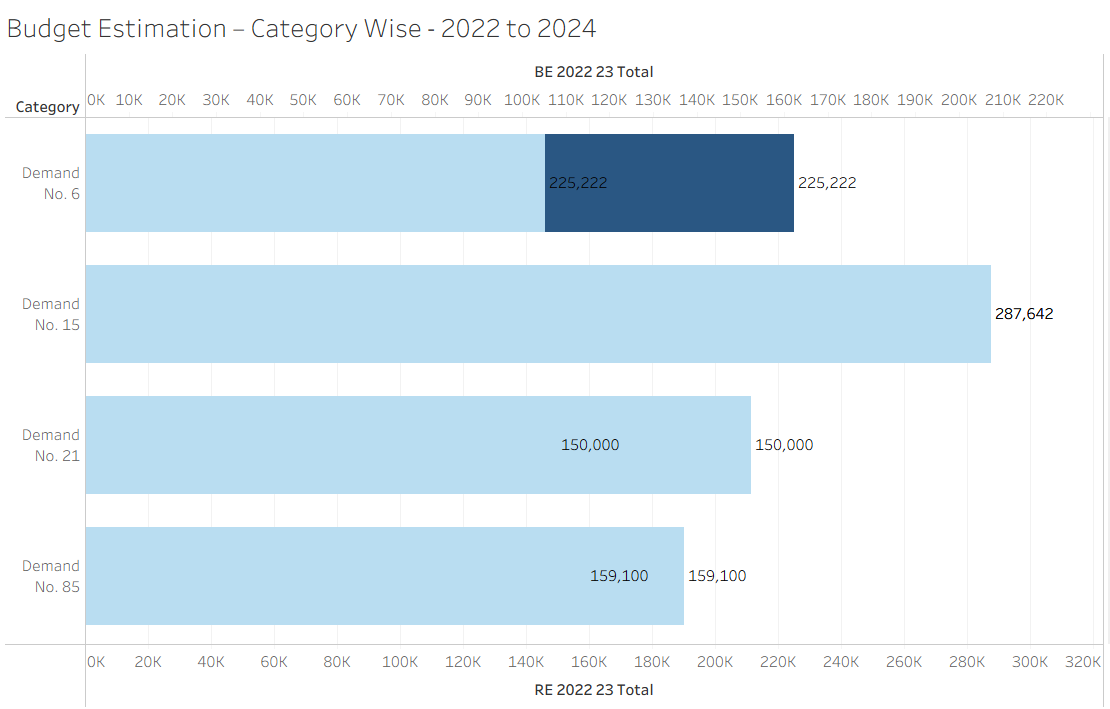
**7. RESULTS**

**7.1 Output Screenshots**

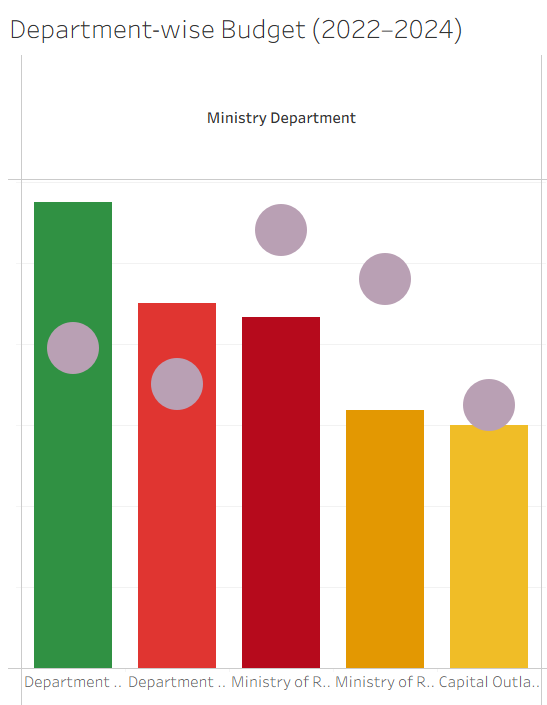
**Activity 1.1: Top 5 Schemes 2021-2022**

****

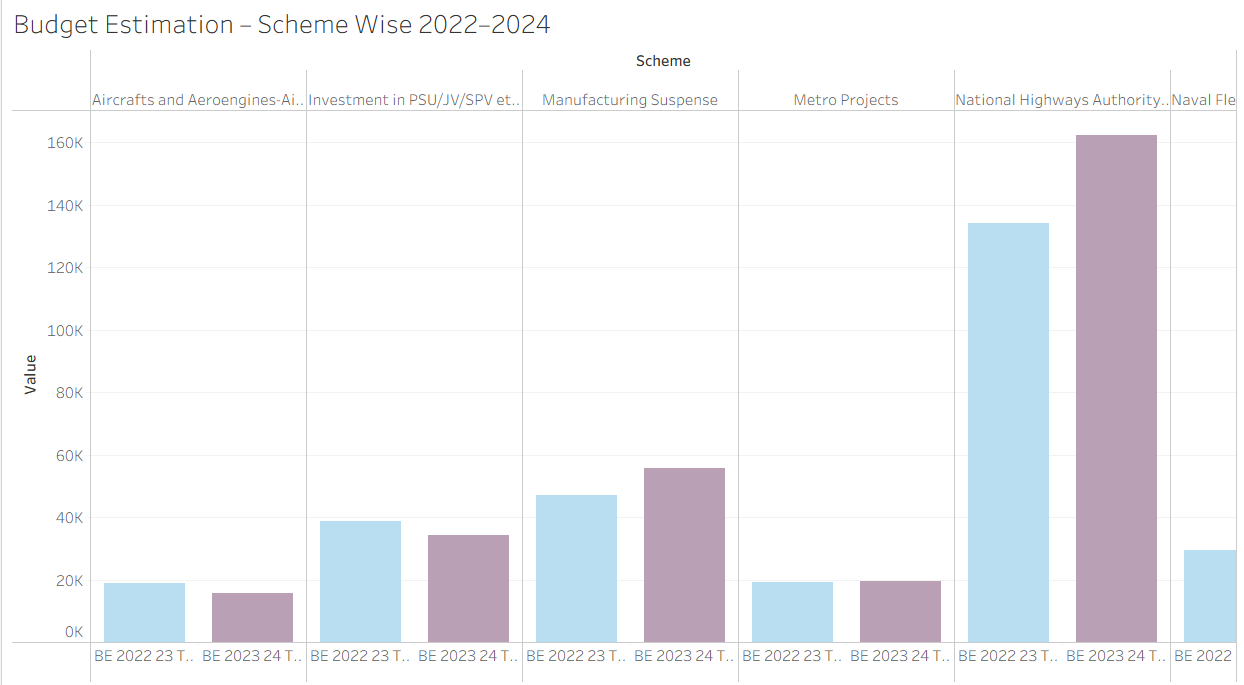
**Activity 1.2: 2022 – 2024 – Budget Estimation – Category Wise**

****

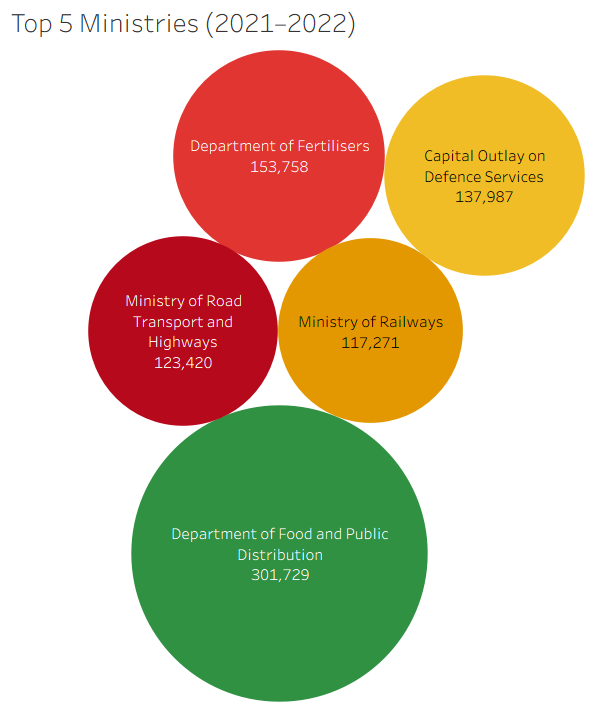
**Activity 1.3: 2022 – 2024 – Budget Estimation – Department Wise**

****

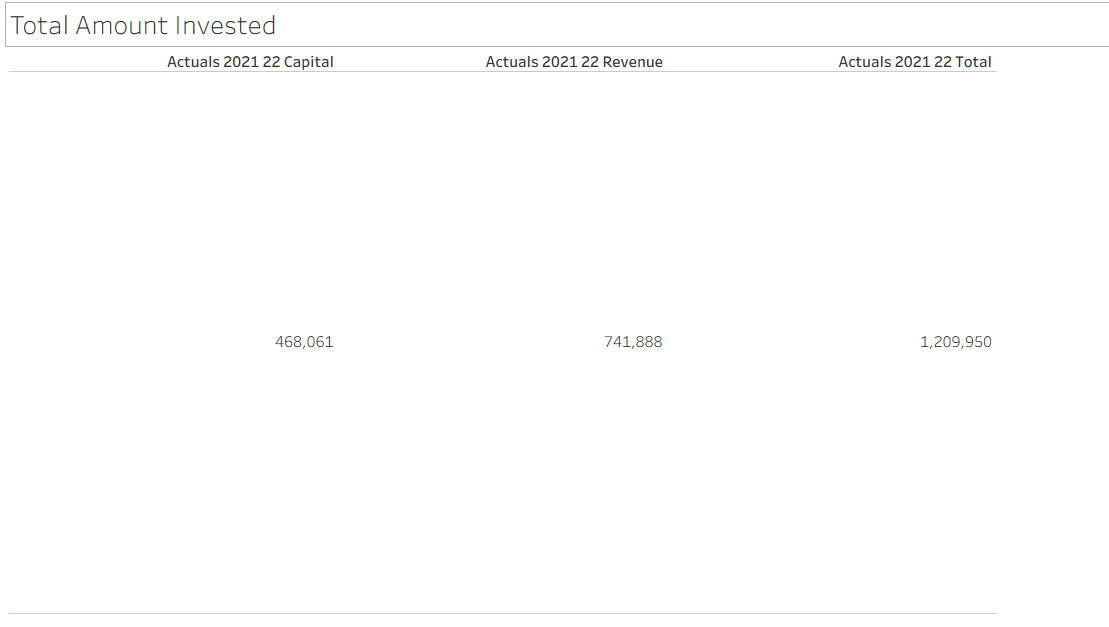
**Activity 1.4 : 2022 – 2024 – Budget Estimation – Scheme Wise**

****

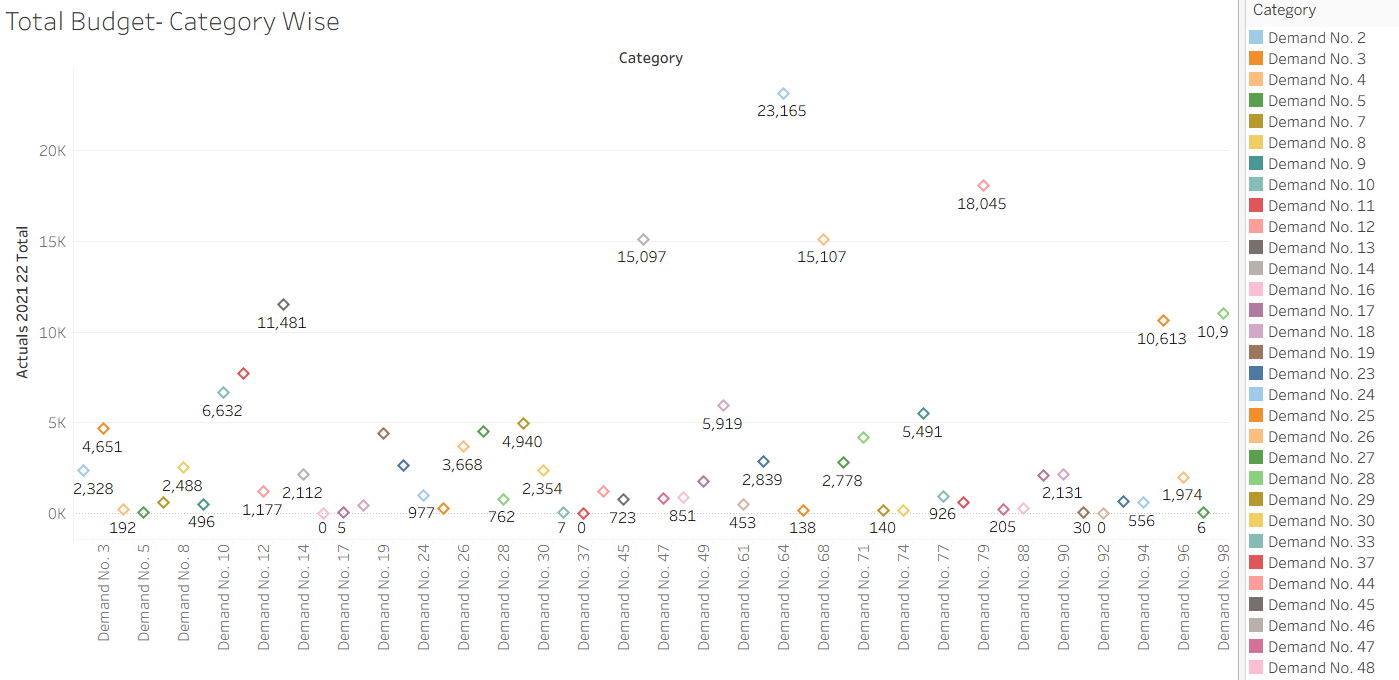
**Activity 1.5 : Top 5 Department/Ministry Wise for 2021-2022**

****

**Activity 1.6 : Total Amount Invested**

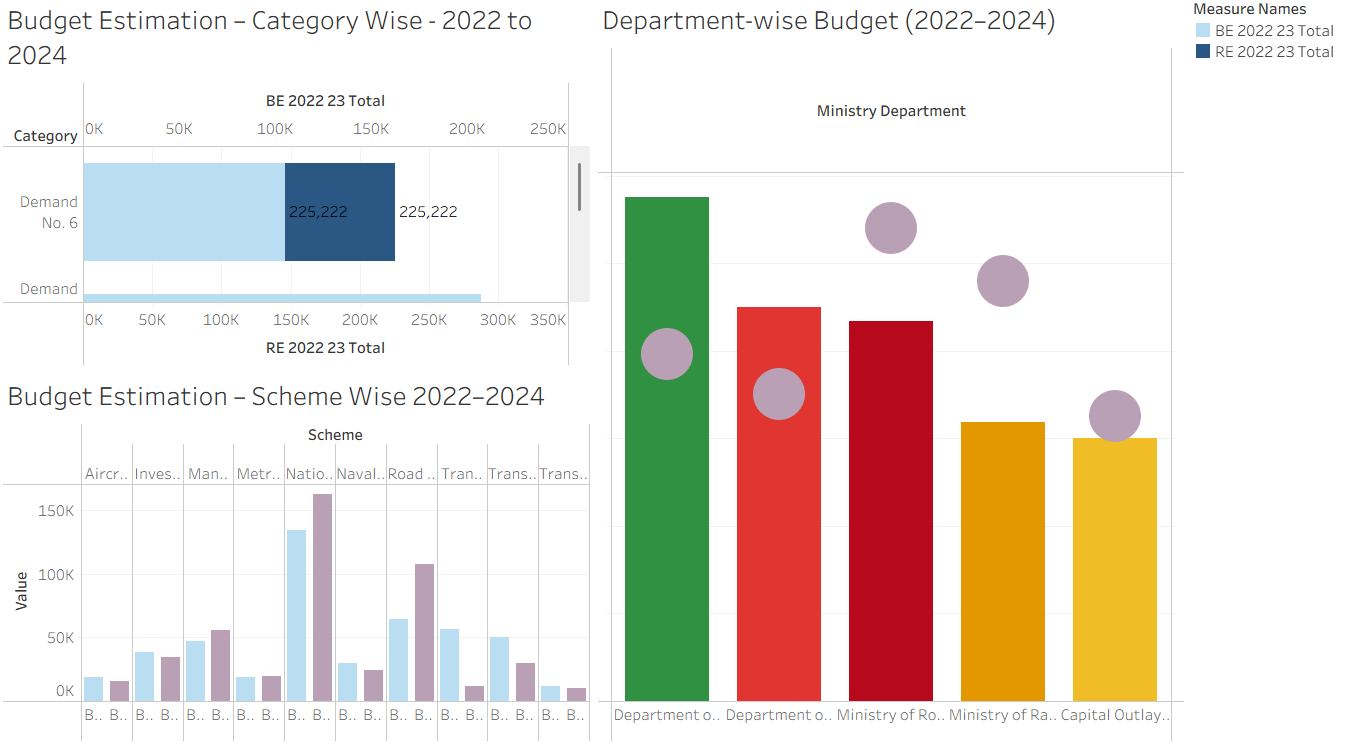
****

**Activity 1.7 :  Total Budget- Category Wise**

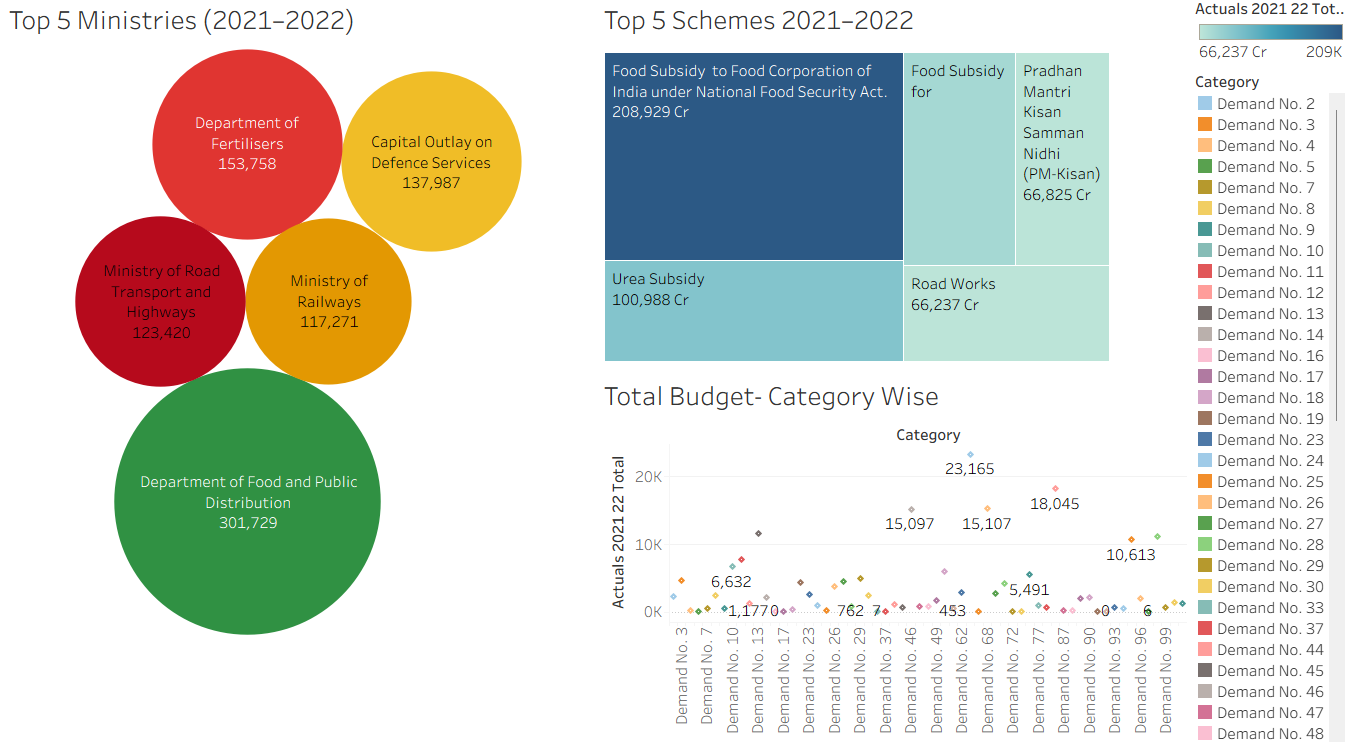
****

**Responsive and Design of Dashboard :**

**Dashboard-1 :**

****

**Dashboard-2:**

****

**8. ADVANTAGES & DISADVANTAGES**

**Advantages**

* Enhances transparency
* Easy year-wise comparison
* Interactive analysis
* Supports data-driven policy insights

**Disadvantages**

* Depends on dataset accuracy
* Requires Tableau knowledge
* Static historical data only

**9. CONCLUSION**

The Union Budget Analysis project demonstrates how data visualization can simplify complex financial information. By leveraging Tableau dashboards and Flask integration, the project transforms static budget data into interactive insights, supporting transparency, accountability, and informed decision-making.

**10. FUTURE SCOPE**

* Add predictive trend analysis
* Include state budget comparisons
* Integrate real-time government APIs
* Build mobile-friendly version
* Add AI-driven financial forecasting

**11. Deployment of Flask Web Application with Embedded Tableau Dashboard**

**11.1 Overview**

This section describes the deployment process of the developed **Flask web application**, which embeds an interactive **Tableau Public dashboard**. The application presents insights from *A College Food Choices Case Study* and has been hosted using **Render.com**, a cloud platform well-suited for deploying Python web services

**11.2 Hosting Platform**

* **Platform:** Render.com
* **URL:** <https://render.com>
* **Purpose:** To host the Flask application on a publicly accessible URL without requiring complex DevOps setup.
* **Reason for Selection:** Render provides free-tier services, native support for Python/Flask apps, easy GitHub integration, and automatic builds.

**11.3 Project Structure**

The Flask application was structured as follows:

|  |
| --- |
| /flask  ├── app.py # Main Flask application logic  ├── requirements.txt # Project dependencies for deployment  ├── Procfile # Specifies how to run the app using Gunicorn  ├── templates/  │ └── index.html # HTML template embedding the Tableau dashboard  ├── static/ # Optional folder for CSS/JS or static assets |

**11.4 Key Configuration Files**

**11.4.1 requirements.txt**

Defines the Python dependencies required by the project. This file ensures Render installs the correct packages during deployment.

|  |
| --- |
| Flask==2.3.2  gunicorn==21.2.0 |

**11.4.2 Procfile**

Instructs the Render platform to launch the Flask app using Gunicorn (a production-ready WSGI server).

|  |
| --- |
| web: gunicorn app:app |

*Note:* app:app refers to the filename (app.py) and the Flask instance (app).

**11.5 Deployment Process**

The following steps were followed to deploy the application:

1. **Repository Setup**
2. The Flask project was uploaded to a public GitHub repository:  
   🔗 [github link for flaskapp](https://github.com/Prithvi2004/Smart-Bridge-Tableau-Intership.git)
3. **Connecting to Render**
   * Logged into Render using GitHub credentials.
   * Selected "New Web Service" and connected the repository.
4. **Configuration Settings**
   * **Build Command:** pip install -r requirements.txt
   * **Start Command:** gunicorn app:app
   * **Runtime Environment:** Python 3 (auto-detected)
5. **Automatic Build & Deployment**
   * Render cloned the repository, installed dependencies, and launched the Flask app.
   * A public URL was generated for accessing the live application.

**11.6 Issue Encountered and Resolution**

During the initial deployment, the following error occurred:

|  |
| --- |
| ERROR: Could not open requirements file: [Errno 2] No such file or directory: 'requirements.txt' |

**Cause:** The requirements.txt file was missing from the repository.

**Resolution:**  
The file was manually created with the appropriate dependencies, committed, and pushed to the GitHub repository. After re-triggering the deployment, the issue was resolved and the application deployed successfully.

**11.13 Final Result**

Once deployed, the Flask application successfully rendered the embedded Tableau dashboard, allowing users to interactively explore the food and nutrition data collected as part of the case study.

deployed URL: [Union Budget Insights Dashboard](https://smart-bridge-tableau-intership.onrender.com/) ( wait 2 mins loading takes time)

**11.8 Conclusion**

The deployment process illustrates a streamlined approach to hosting data visualizations through Flask and Tableau using Render. This solution enables the delivery of dynamic dashboards to end-users via a lightweight, scalable, and cost-effective platform.

**All Links**

**Tableau Viz Public URL :** [Union Budget](https://public.tableau.com/views/UnionBudget_17711331782220/Dashboard1?:embed=yes&:showVizHome=no)

**GitHub link for flask :** [github link](https://github.com/Prithvi2004/Smart-Bridge-Tableau-Intership.git)

**Project Demo Live link :** [Union Budget Insights Dashboard](https://smart-bridge-tableau-intership.onrender.com/) ( wait 2 mins loading takes time)