UI/UX Design Documentation of Treelytics

Project Overview

Project Name: Treelytics

Project Description

Treelytics is a data analytics app which collects raw data from various sources such as CCTV camera, Live video, CSV and so on then processes it further to present the data in the form of different comprehensive data visualisation elements. The data visualisation elements consist of data card, bar graphs, pie charts, bar-lines, time series, treemap, area, geo-chart, heatmap which aid in user data comprehension.

Objectives

- It seamlessly integrates diverse data sources, including CCTV cameras, live video feeds, and CSV files.
- It enhances user comprehension of complex data through a variety of data visualisation elements.
- It improves decision-making by providing clear and accurate visual representations of data.
- It designs an intuitive and accessible interface that caters to users of all technical backgrounds.
- It enables real-time processing and updating of data visualisations to provide the most current information.
- It allows users to customise their data visualisation dashboards according to their specific needs and preferences.
- It ensures the app can handle large volumes of data while maintaining high performance and responsiveness.

Stakeholders

- **Product Manager:** Ashok Kumar Pant, Dipak Malla
- **Lead Developer:** Bikash Saud
- **UI/UX Designer:** Jenisha Khadka

1. Design Principles

- **User-Centred Design Principles:**

- **Focus on user Needs:** The design prioritises ease of use for users with varying technical backgrounds by ensuring that the interface is intuitive and straightforward.
- **Intuitive Navigation:** The navigation structure is simple and logical, allowing users to find and analyse data efficiently without a steep learning curve.
- **Actionable Insights:** Data visualisation elements are designed to provide clear and actionable insights, helping users to make informed decisions quickly.

- **Accessibility Considerations:**

- **High contrast:** Ensured high contrast between text and background to improve readability for users with visual impairments.
- **Charts and Graphs:** Each data visualisation element, such as bar-graphs and pie-charts, includes a detailed description

- **Visual Design Guidelines:**

- **Consistent Design Language:** Maintained a consistent design language across the platform to ensure familiarity and ease of use. This includes consistent use of colours, typography, and iconography.
- **Natural Colour:** Utilised a shade of white for the background to create a simple and clean look. The primary colour, blue, is selectively used in buttons to draw attention to key actions without overwhelming the user.
- **Enhancing Usability:** All visual elements are designed to enhance usability and clarity. For example, data visualisations like bar graphs, pie charts, and heatmaps are chosen for their ability to convey complex information in an easily understandable manner.
- **12-Column Grid Layout:** Employed a 12-column grid layout to ensure a structured and balanced design. This includes margins, gutters, and responsive adjustments to maintain a clean and organised interface across different devices.

2. Design Specifications

a. Colour Palette:

- **Primary Colour:** #0F52BA
- **Secondary Colour:** #FFFFFF
- **Accent Colour:** #949494
- **Background Colour:** #FAFAFA
- **Text Colour:** #272727

b. Typography:

- **Headings:** Roboto, Regular, 30px
- **Sub Headings:** Roboto, Regular, 25px
- **Body Text:** Roboto, Regular, 16px

c. Iconography:

- Icons used here are of 24px by 24px and colour #4E4E4E

d. Spacing and Layout Guidelines:

- **Grid Systems:**
  - **Columns:** 12 columns
  - **Type:** Stretch
  - **Margin:** 150 px
  - **Gutter:** 24 px
  - **Width:** Auto

3. Component Library

a. Buttons:

- **Primary Button:** Button colour used #0F52BA with text colour #FEFEFE
- **Secondary Button:** Button colour #0F52BA with stroke and text colour #FEFEFE

- **Disabled Button:** Button colour #0F52BA with stroke #949494 with text colour #FEFEFE

b. Forms:

- **Single text input:** Roboto, Regular, 16px with stroke of colour #F0F0F0 of size 1.25.

c. Modals:

- **Consistency:** Modals are designed uniformly across all projects to maintain consistency.
- **Position:** Modals are centred on the screen to ensure they capture the user's focus.
- **Overlay:** A clear overlay is used to darken the background, preventing interaction with the main content until the modal is dismissed.
- **Content layout:** The modal content is structured with a title at the top, followed by detailed information presented in a clear and organised manner.
- **Close Button:** Each modal includes a close button (typically an 'X' in the top right corner) for easy dismissal.
- **Actions:** Modals include primary actions such as Add, Save, Upload, Create, Next, Yes and OK and secondary actions such as Cancel, No and Reset to guide the user through the necessary steps.
- **Size:** Each modal is of different size of width x height as (350x271, 526x217, 527x183, 606x225, 615x198, 615x235, 615x230, 680x445, 680x504, 680x256, 680x310, 698x220, 725x340, 790x340, 790x670, 792x560, 790x350, 900x605, 1350x735 and 1489x872)

d. Progress Card:

- **Purpose:** Progress cards display the status of ongoing tasks. Such as file processing.
- **Content Layout:** Includes a title (e.g., Processing files), individual progress bars for each task, and status indicators.
- **Visual Indicators:** Use progress bars to visually represent the percentage of completion for each task.

- **Interactivity:** It includes options to expand/collapse the card to view more or less detail.
- **Positioning:** Positioned in the bottom right corner of the screen for easy visibility without obstructing the main content.
- **Updates:** Dynamically updates to reflect the current status of each task.

4. Main Pages

a. **Dashboard:** This is the first page user sees which consists of two buttons to '**create dashboard**' and '**add data source**' and favourite dashboard made by the user along with a list of dashboards.

   After the user clicks the '**create dashboard**' a page opens which consists of **edit** and **view** mode along with all data controls and data visualisation elements like data card, bar graphs, pie charts, bar-lines, time series, treemap, area, geo-chart, heatmap which aid in user data comprehension.

   In **edit mode**, a blank layout is opened with control section at top and side panel with **DATA** and **STYLE** section.

   In the **DATA** section, there is a search bar to search for queries and below it has a list of queries through which **users can select a certain query** and **get suggestions for the chart** which is shown on the layout.

   In the **STYLE** section, the user can view the chart and change its title,legend name and they can also change the visibility of the legend and its placement.

In **view mode,** the user can see the visual representation of the data available on the page.

The user can **share**, make **favourites** and **save** the dashboard.

The user can **zoom in** as well as **zoom out** the data in the dashboard.

The user can **add different charts, tables, images, texts, lines, shapes** in the page.

Also the user can **choose layout size** for the dashboard.

For the '**add controls**' if there is no visual data on the page then it is disabled. After adding visual data the user can implement the add controls filter.

b. Data Source: Initially in the data source page a layout with different data sources (**Camera**, **MySQL**, **CSV**) are available.

A specific page opens up when the user clicks each data source.
When the user selects **Camera source** a new page opens for Camera configuration including different input fields such as Camera Name, Source URL, Location and a dropdown for selecting features and a link for Region of Interest.

Secondly when the user selects **MySQL** data source a new page opens for MySQL configuration including different input fields such as Host, Port, Source Name, Database Name, Username, Password, URL. Here Host, Port and Source Name are required fields.

The ‘**Test connection**’ button here is to make connections and to import files.

If the connection fails then the data cannot be imported. Similarly when a user selects CSV data source, a new page opens for CSV configuration including input fields such as Source Name and URL.

Users can either put the URL in the URL field or upload a CSV file.

When uploading a CSV file a new page opens for importing data for CSV file which shows different dropdown button with labels such as Date Format, Test Qualifier, Delimiter for separating columns, Thousand Separator, Decimal Separator and toggle button whether or not to show the first column name. And an increment/decrement button to skip the first row. After adding specific data sources, it is shown in the data source page as table view.

Here in the table only the **Camera source file** is shown with a toggle button which is when toggled on, the data from a given CCTV camera is imported and is not imported when toggled off.

The user can also **view** the details, **edit** and **delete** the data sources.

c. **Data Table:** Initially the data table page shows a search bar along with a button as ‘**Add Data Table**’ and a table view of data as Table Name, #Count, Size, Data Source and Created Time.

In the **Data Source** a label is shown as ‘**streaming**’ which means . A detailed view of the data table is shown when the user clicks on specific data.

When the user clicks on **add data table** a pop up shows up with input fields such as **Data Source** and **Table Name** with both fields required.

Similarly a popup for validation error popups when a user enters duplicate data.

After entering the required fields a popup opens up where we can select different schemas (**Camera**, **MySQL** and **CSV**).

There is a table below with header SN, Name, Attribute, Type, Transform, Default Value, Not Null, Unique, Auto Increment. After selecting the default schema and clicking on the apply button the input fields in the table are auto filled.

We can also **add columns** in the table.

The user cannot use the same name for the table name as the name must be unique.

After completing all the processes the data is updated in the **Data Table page**.

There might be many files yet to be shown in the page so we can track the progress of the files being updated in the processing files layout on the button-right of the page.

d. **Query Builder:** Initially the query builder page shows a search bar along with a button for '**Add Query**' and list of queries with headers such as SN, Query Name, Queries, Created date.

The user can **delete** the data from the table as well as **edit** the query they can simply click on one of the queries which will open a new page for editing queries which consists of an input field named 'Query name' and an input box for query builder where users can write their queries.

The query builder has two modes as '**Code**' and '**Builder**'.

In the **Code mode**, the user will manually write the SQL code in the input box and after testing the query, data will be shown in a tabular format.

While in the **Builder mode**, the users will have a GUI to import the queries. In the Builder mode, the users can use components to write a query, they are provided with features as Filter, Order by, Group by and Preview.

The user must select a data source, data table, column name, (aggregation and alias) which are optional.

The user can also **create new columns** and **join tables**.

There is an icon on the top right corner for the query prompt when clicked, a sidebar opens up in which the user can select multiple tables, after selecting the table, the user can write the prompt and the query will be generated by the query prompt.

e. **Settings Page**

Overview:
This section provides details on the settings page of the web application, designed to manage user profiles and roles efficiently. The settings page is divided into two main sections: the user profile and user management.

- **Navigation Bar:**
  Location: Top of the page.
  Contents: Home, Dashboard, Data Source, Data Table, Query Builder, Settings, Alert.
  Function: Allows users to navigate between different sections of the application.

- **Sidebar Menu:**
  Location: Left side of the page.
  Contents: General, Profile, Manage User, Manage Role, and redundant "Home" entries.
  Highlight: The active section is indicated with a blue background.
- **Profile Section:**
  Components:
  User Profile Card: Displays the user's profile picture, name (Adam Magar), email, role, and status.

Edit Button: Positioned on the profile card for easy access to edit profile information.
Design: Clean and organised layout for quick access and modification of profile details.

- **Manage User Section:**
  Components:
  Search Bar: Positioned at the top for quick user search capabilities.
  User List Table:
  Columns: User (with profile picture), Role, Status.
  Status Indicators: Text (Active/Inactive) and toggle switches for status changes.
  "Add User" Button: Positioned for quick addition of new users.
  Pagination Control: At the bottom for navigation through multiple pages of user data.
  Visual Hierarchy: Consistent design elements, ample white space, and clear section separations enhance readability and usability.
  Potential Improvements:

  Remove redundant "Home" entries to streamline navigation.
  Add role descriptions or permission levels for more context in role management.
  Include more profile details such as last login time, department for a more informative profile section.
  Implement confirmation dialogs and notifications for critical actions and success/failure feedback.

f. Alert Page

Overview: This section describes the alerting page of the web application, designed to manage and create alert rules efficiently. The alerting page is divided into two main sections: the alert list and the add alert page.

- **Navigation Bar:**
    - **Location:** Top of the page.
    - **Contents:** Home, Dashboard, Data Source, Data Table, Query Builder, Settings, Alert.
    - **Function:** Consistent and accessible navigation across the application.
- **Sidebar Menu:**
    - **Location:** Left side of the page.
    - **Contents:** Alerting, Alert List, Add Alert.
    - **Highlight:** The active section is indicated with a blue background.
- **Alert List Page:**
    - **Components:**
        - **Search Bar:** Positioned at the top for easy searching of specific alerts.
        - **Alert List Table:**
            - Columns: Alert, Query Name, Created Date, Period, Status.
            - Status Indicators: Toggle switches for activating or deactivating alerts.
            - "Add Alert Rule" Button: Prominently displayed for quick addition of new alerts.
    - **Visual Hierarchy:** Clear separation of elements, appropriate white space, and primary actions highlighted in blue.

- **Add Alert Page:**

- **Form Layout:**
    - **Set Alert Name:** Text input for the alert name (required field).
    - **Select Query:** Dropdown for selecting a query, text area for displaying the query, and a "Test Query" button for immediate testing.
    - **Alerting Condition:** Dropdown for condition selection and text input for threshold value.
    - **Alerting Frequency:** Dropdown for selecting the frequency of alerts.
    - **Notification:** Checkbox for email notifications, default checked state.
- **Visual Hierarchy:** Numbered steps guide users through the form logically, with clear labels and input fields.

Potential Improvements:

- **Add descriptive information or icons** for each alert in the alert list for more context.
- **Include filtering options** in addition to the search bar for managing numerous alerts.
- **Provide tooltips or help icons** next to each input field in the add alert form for user assistance.
- **Implement inline validation** to inform users of errors or missing information during form filling.

Reference

Lookerstudio - Looker Studio (Mostly Used Reference)

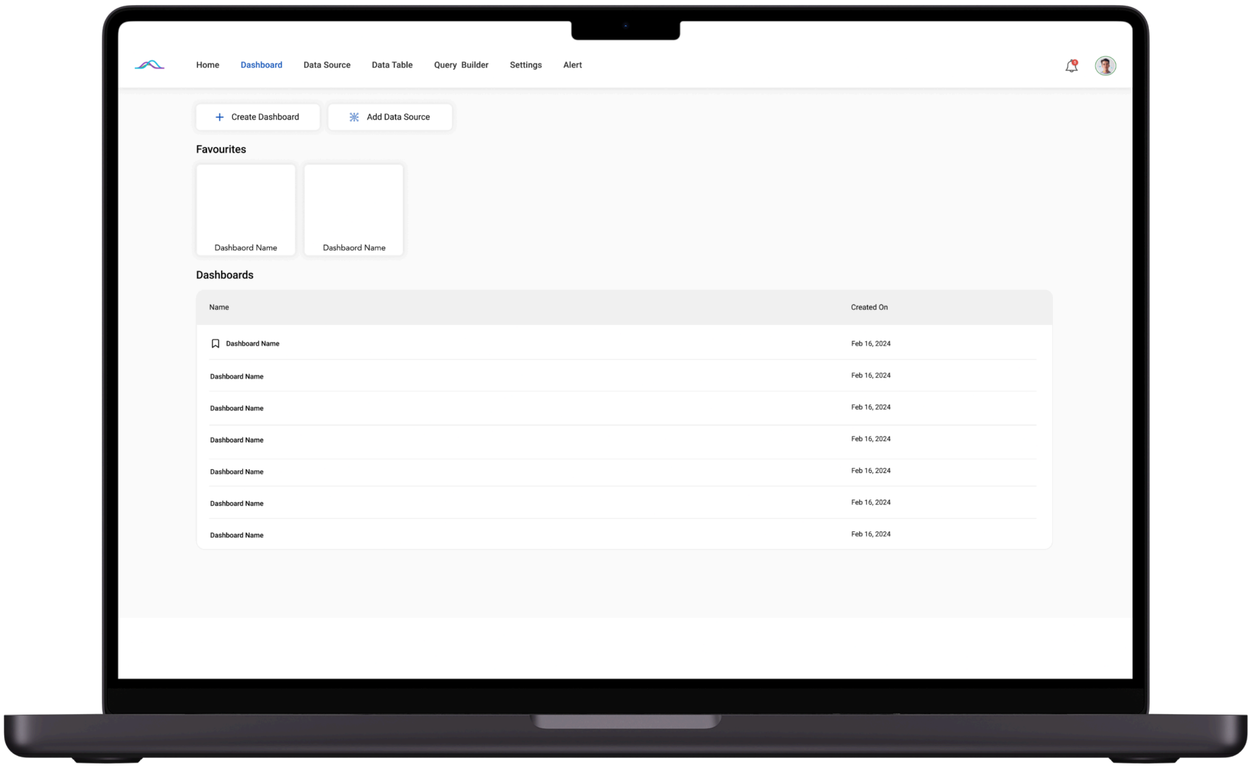
Grafana - Grafana

Qlik - Qlik

Tableau - Tableau

Zoho - **Zoho**

Laptop Mockup:



Favourites

Dashbaord Name

Dashbaord Name

Dashboards

Name	Created On
Dashboard Name	Feb 16, 2024
Dashboard Name	Feb 16, 2024
Dashboard Name	Feb 16, 2024
Dashboard Name	Feb 16, 2024
Dashboard Name	Feb 16, 2024
Dashboard Name	Feb 16, 2024
Dashboard Name	Feb 16, 2024