System Design Architecture

1. Frontend (User Interface):

- **Streamlit Framework:**
    - The user interface is built using Streamlit, a popular open-source framework for creating data-driven web applications.
    - Streamlit provides the structure and components for building the UI, such as headings, sidebars, select boxes, and tables.
- **Components:**
    - **Sidebar Navigation:**
        - Allows users to navigate between different sections of the application.
    - **Content Sections:**
        - Each section displays different IPL statistics, such as batsman statistics, bowler statistics, head-to-head matches, etc.

2. Backend (Data Processing and Logic):

- **Data Loading:**
    - A CSV file `deliveries.csv` is loaded using the `pandas` library.
    - The `load_data` function, cached by `@st.cache` to optimize performance, reads the CSV file and returns a DataFrame.
- **Data Processing Functions:**
    - Various functions process the IPL data to calculate specific statistics:
        - `calculate_batsman_stats`: Calculates statistics like total runs, sixes, fours, matches played, half-centuries, centuries, batting average, strike rate, and highest score for each batsman.
        - `calculate_bowler_stats`: Calculates statistics like runs conceded, balls bowled, wickets taken, economy rate, and five-wicket hauls for each bowler.
        - `batsman_with_most_runs`: Determines batsmen with the most runs.
        - `bowler_with_most_wickets`: Determines bowlers with the most wickets.
        - `batsman_with_most_sixes`: Determines batsmen with the most sixes.
        - `batsman_with_most_fours`: Determines batsmen with the most fours.
        - `calculate_bowler_vs_batsman_stats`: Calculates statistics for a selected batsman against a selected bowler.
        - `head_to_head_match_details`: Calculates head-to-head match details between two selected teams.
- **Data Caching:**
    - The use of `@st.cache` ensures that the data is loaded and processed once, and reused across multiple runs, improving the application's performance.

3. Integration (Frontend-Backend Interaction):

- **Data Binding:**
  - The frontend components (e.g., select boxes, tables) interact with the backend functions to fetch and display data based on user inputs.
- **Interactivity:**
  - Users select options from the sidebar to view specific statistics.
  - The application updates the displayed data dynamically based on user selections.

4. Visualization:

- **Tabular Data:**
  - Statistics are presented in tabular format for easy readability.
- **Styling and Layout:**
  - The layout is customized using Streamlit's `set_page_config` and CSS styling for a better user experience.
  - A background image is included to enhance the visual appeal.