International Journal on Science and Technology (IJSAT)
E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Next-Gen Dashboards with Oracle APEX: From Charts to Smart Filters

Ashraf Syed

maverick.ashraf@gmail.com

Abstract:
Oracle Application Express (APEX) is a low-code development platform tightly integrated with Oracle Database. APEX provides powerful next-gen dashboard with smart interactive capabilities, combining rich charting, data filtering, and modern UI design. This article surveys APEX dashboard features, emphasizing data visualization options (chart types, stacking, drill-down links), UI/UX enhancements (Universal Theme, template directives, faceted search, smart filters), and data integration (RESTful and external sources). The article enumerate supported chart types (bar, line, area, pie, donut, funnel, gauge, scatter, bubble, heatmap, gauge, stock, and more) and explain how to create stacked bar charts and enable drill- down from charts to detail pages. A novel approach is proposed that leverages APEX's template components and dynamic actions to improve dashboard usability. Code snippets illustrate building chart regions and filters (e.g. SQL queries for charts and JavaScript API calls), and screenshots show example dashboards. The intended audience includes database professionals, business analysts, researchers, and students interested in practical and advanced APEX dashboard design. This review references official APEX documentation and credible sources to give a complete, current picture of APEX dashboard capabilities.

Keywords: Oracle APEX, Interactive Dashboard, Data Visualization, Oracle JET Charts, Universal Theme, Template Components, Faceted Search, Smart Filters, REST Data Source, Drill-Down, Stacked Bar Chart, UI/ UX, Low-Code, Data Integration.

I. INTRODUCTION

Interactive dashboards have become indispensable tools in modern data-driven environments. They provide intuitive platforms for visualizing complex datasets and supporting real-time monitoring of key metrics, enabling data-driven decision making. By centralizing charts, tables, and filters on one screen, dashboards help reduce information overload and improve situational awareness. They are used across domains from business intelligence and healthcare to education, manufacturing, and more [1]. Dashboards facilitate tasks like tracking KPIs, anomaly detection, and exploratory analysis. Research emphasizes that dashboards must balance information richness with usability, since overly complex displays can overwhelm users [2].

Oracle Application Express (APEX) is a rapid web-app development platform built into Oracle Database. It enables developers to build enterprise web applications with minimal coding by using wizards and declarative components [3]. APEX's Universal Theme provides a responsive, modern UI framework out-of- the-box, allowing the creation of rich interfaces without extensive web design expertise [3]. In particular, APEX has extensive support for creating interactive dashboards that combine charts, reports, and filters. These dashboards are widely used for analytics and reporting by database administrators, business analysts, and educators.

An APEX interactive next-gen dashboard is a page (or set of pages) that presents data visually through charts and reports, and allows end-users to explore the data by filtering, grouping, or drilling down. For example, a sales dashboard might show bar charts of revenue by region and pie charts of product mix, with controls to drill into regions or filter by time period. In APEX, dashboards can contain Interactive Reports/Grids, Classic

Reports, calendars, faceted searches, and chart regions. A key strength is APEX's ability to integrate multiple data sources (local tables, REST services, SaaS data) and custom actions (Dynamic Actions, plugin components) into one page. This makes it suitable for rapid BI application development [4] [5].
APEX charts are built on Oracle JET (JavaScript Extension Toolkit) visualizations, providing dozens of chart types and interactive features [6]. Standard chart types include bar (vertical/horizontal), line, area, scatter, bubble, polar, radar, pie, donut, funnel, stock, and more [6]. Charts can be multi-series or single-series, with options for stacking, dual axes, formatting, and live updates. Charts are connected to SQL queries or REST sources, so they display real-time data. APEX also supports embedding AnyChart charts (legacy) and plugin-based regions, but the recommended approach is Oracle JET for new apps.

Key usability features in APEX include faceted search regions and Smart Filters for quick, interactive filtering without writing code [5] [7] . The latest release adds enhancements like more flexible facets (arbitrary operators, descending sort) and streamlined theme components [5]. It also updated the built-in JavaScript libraries (Oracle JET upgraded to 15.0.0) and introduced new page items (e.g. combobox, image upload, QR code) [5].
APEX allows dashboard regions to fetch data from local tables, REST APIs (including ORDS, OData, and Oracle Cloud SaaS) and Web Source Modules (generic JSON services) [4]. This means dashboards can combine internal and external data – for example, blending on-premise sales data with cloud analytics or third-party feeds. The platform provides a REST Data Sources framework with credentials and schema mapping, enabling secure, declarative integration [4]. Developers can also use Web Source Modules to consume generic JSON and CSV feeds.

In summary, an Oracle APEX dashboard leverages low-code components (reports, charts, filters, plugins) to deliver a rich, interactive analytic interface. By 24.2, APEX has matured into a versatile BI toolkit. This article examines in detail how to implement interactive dashboards in APEX: the types of charts available, methods for creating stacked and drillable charts, UI/UX best practices (themes, components, filters), and integrating diverse data. Author also proposes a novel approach to improve dashboard design via APEX's Template Components and advanced JavaScript APIs, and outline future directions (e.g., AI integration, augmented analytics).

II. BACKGROUND AND LITERATURE REVIEW

Interactive dashboards have become essential in modern analytics and business intelligence. A dashboard typically aggregates multiple visualizations and tables on one screen, allowing users to compare metrics and explore data. According to Oracle's design materials, "interactive dashboards" let end-users manipulate data views directly (for example, by filtering or sorting on charts) for instant insight [4]. In APEX, the original Interactive Dashboard pages (introduced in earlier versions) enabled users to add multiple "analyses" (IRs, charts, lists) and link them via parameters. Over time, APEX has evolved its dashboard paradigm to include Faceted Search and Smart Filter regions (since version 21.1), and to integrate enterprise data sources via REST. Oracle's documentation highlights that the APEX App Builder supports a wide variety of chart types, which cover most common BI needs [6]. These include basic charts (bar, line, pie) as well as more specialized ones (donut, funnel, stock/financial, polar, radar). For example, the App Builder User's Guide lists "bar, line, area, range, combination, scatter, bubble, polar, radar, pie, donut, funnel, and stock charts" as supported types [6]. The official APEX Sample Charts application provides live examples: it demonstrates how bar charts can use series colors, dual axes, stacking, percent-stack, and custom legends [8]. In practice, developers often use Bar, Column (vertical/horizontal) and Line charts to show categorical trends, pie/donut charts for part-to-whole comparisons, and Scatter or Bubble charts for correlation analysis. The JET chart set also includes Gauge and Heat Map components, although some require special configuration.

Stacking is a frequently used chart feature for comparing contributions of sub-categories. APEX supports stacked bar charts declaratively: one can simply enable the "Stack" toggle on a bar chart's attributes to stack series together. A sample stacked bar chart displays percentage stacking by setting the axis format to Percent, automatically normalizing values. The literature on data viz suggests that stacking clarifies part- to-whole composition, although it can obscure individual trends if too many series are stacked. Therefore, interactive APEX dashboards often include controls (e.g., checkboxes or buttons) to toggle stacking on/off. For example, Adjirackor (2024) demonstrates using buttons and dynamic actions to switch a bar chart between horizontal/vertical orientation and stacked/unstacked modes. In his example, a chart region (ID ownaCarChart) initially shows data grouped by age and car ownership status with stacking enabled.
Dynamic Actions call the JET API to reconfigure the chart on-the-fly:

```
apex.region('ownaCarChart').widget().ojChart(
        {orientation: 'horizontal'});
apex.region('ownaCarChart').widget().ojChart(
         {orientation: 'vertical'});
apex.region('ownaCarChart').widget().ojChart(
               {stack: 'off'});
apex.region('ownaCarChart').widget().ojChart(
                {stack: 'on'});
```

These commands (which leverage widget().ojChart(...) on the chart region) instantly transform the chart without a full page reload [9]. This approach shows the flexibility of APEX's client-side API: developers can bind buttons to such actions, giving end-users UI controls for the chart's appearance.
Drill-down and interactivity are also hallmarks of dashboards. In APEX, a chart or report element can contain links to detailed pages or reports. For charts, APEX allows defining a Link attribute at the chart or series level. For example, the "Series Name Column Mapping" example in the Oracle sample uses a SQL query with a URL-building column. Each chart element's click action then redirects to another page (e.g. an Interactive Grid) with filter parameters passed. As documented, clicking a bar segment can redirect to Page 29 of the application, with that page containing an IG that filters its DEPTNO column based on the clicked bar's department number [8]. The setup is done by setting *Link Type* = Redirect to Page, targeting Page 29, and in *Link Builder* specifying which page item (like IG_DEPTNO) to set from the chart's series value [8]. This declarative linking provides immediate drill-down: if a user clicks "Sales" on the revenue chart, the app opens a report listing each sale in that category. Oracle's older tutorials also describe how classic reports can link to detail forms (drill-down reports) [10]. Modern APEX emphasizes using the new URL and link builder mechanisms for charts and IRs, making drill-through seamless.

Beyond charts, APEX interactive dashboards often include Interactive Reports (IR) and Interactive Grids (IG), which let users re-sort, filter, and pivot data on the fly. For instance, an IR can group rows by a column and then users can drill into each group by clicking its summary row. APEX's enhancements to Faceted Search and Smart Filters greatly simplify adding dashboard filters. Faceted search regions are declarative filter builders: users can click on column values (or type in text) to filter a linked report region. The 23.2 release adds "arbitrary facet" support, meaning text input facets now allow operators like "not equals" or "does not contain" [5]. It also lets developers preset sort order for LOV facets. Smart Filters (introduced in 22.2 version) embed a search bar that can apply filters by typing or suggestions. These features provide out-of- the-box, intuitive filtering experiences without coding. From a UI/UX perspective, APEX uses the Universal Theme to achieve responsive, accessible design. Universal Theme follows UI best practices (cards, modals, navbars, etc.) and is touch-friendly [3]. All region and item templates can be customized via Template Options (for color, alignment, icons) at design time or via the Theme Roller at runtime. The 24.2 release further improved the theme system: for example, Template Directives in Static Content (such as {with ...}/{/with} blocks) now support more attributes, enabling advanced templating within pages [5]. There is also an updated "Redwood Light" theme (Oracle's new design language)

with refined color contrasts and spacing [5]. These UI advances ensure dashboards are not only functional but visually appealing and consistent with enterprise branding.

On the data integration front, APEX has comprehensive support for external data. The built-in REST Data Sources feature lets developers define named REST endpoints and use them in classic/interactive reports, calendars, and JET charts [4]. Types of REST sources include generic HTTP/JSON feeds, Oracle ORDS endpoints, and the new Oracle Cloud SaaS/OCI REST connector (which supports OData) [4]. This means one can connect a dashboard region directly to, say, Oracle Fusion Cloud ERP data or a third-party JSON API, and render that in charts or reports. APEX even supports REST Enabled SQL sources: these are REST endpoints wrapping a SQL query, which can be refreshed locally for performance [4]. In practice, a developer might define a REST Data Source pointing to an HR API; then a SQL Query region uses that source as its data. The data integration is bi-directional too: forms and charts can invoke POST/PUT/DELETE operations on REST sources, enabling live updates.

While there is limited formal research on Oracle APEX dashboards specifically, several research papers, whitepapers and community articles attest to the effectiveness of Interactive Dashboards in general. A study by Myakala et al. emphasize dashboards as "centralized hubs" where diverse data are consolidated into charts, and interactivity (filtering, drill-down) is supported. This study agree that dashboards help monitor performance and support decision-making [11]. At the same time, research by Alhamadi et al. note design challenges: too many charts or controls can overload users and explicitly report that "having a dashboard with too much data can negatively affect decision making" [1A}. Oracle themselves claim APEX is the *"world's most popular enterprise low-code application platform"* used by hundreds of thousands of developers. Indeed, one study notes approximately 500,000 developers already use APEX's browser-based environment to create web and mobile apps [2]. Vogel et al. have noted that faceted filtering became widely adopted in many domains over the last decade. This technique comes from information retrieval: faceted navigation allows users to iteratively narrow results by multiple dimensions [12]. Oracle's own cloud documentation highlights using APEX (or Oracle Analytics Cloud) for quick dashboards on Autonomous Database data [13]. Industry blogs note APEX dashboards are popular in higher education and enterprises due to rapid development and database integration. The anychart.com site and ApexCharts documentation (third-party) list chart usage guidelines, indicating APEX's charts align with general data-viz best practices. The key takeaway is that APEX's dashboard capabilities, combined with its ease of use for Oracle DB projects, make it a compelling BI development platform.

In summary, prior work and documentation show that Oracle APEX provides a rich dashboard toolkit: dozens of chart types [6], interactive reports, and advanced filters. The environment is optimized for building analyzable applications with minimal coding. The novelty in this article will be to synthesize these features in the context of APEX, highlight UI/UX and integration best practices, and propose innovative techniques (e.g. template-component driven dashboards). This paper will cover each aspect—charting, stacking, drill-down, UI frameworks, REST integration—in depth, with code examples and references to documentation and community examples.

III. CHART TYPES AND INTERACTIVE VISUALIZATION IN APEX

Oracle APEX supports an extensive variety of chart types, enabling dashboards to convey different data insights. According to Oracle's App Builder guide, the available JET-based charts include the following (non-exhaustive) list [6] [8]:

- **Bar/Column Charts:** Vertical or horizontal bars; can display single or multi-series data. Variants include stacked bars (stacking one series on another) and percent-stacked (normalized to 100%) [8].
- **Line Charts:** For trends over a continuous axis; can be combined with area fills or multiple series.

- **Area Charts:** Similar to lines but with filled areas, useful for emphasis.
- **Combination Charts:** Mix of lines and bars in one chart.
- **Pie/Donut Charts:** Circular charts for part-to-whole; APEX supports Pie and Donut (ring) styles.
- **Funnel Charts:** For showing stage-wise reduction (e.g. conversion funnels).
- **Radar (Spider) Charts:** For multivariate data on an axis from a center point.
- **Polar Charts:** Similar to radar but with category angular axes and radial values.
- **Scatter/Bubble Charts:** XY plots (scatter) or with size dimension (bubble).
- **Heatmap:** Matrix of values (colors), useful for dense categorical data.
- **Box Plot:** Statistical box-and-whisker charts (underlying data summary).
- **Gantt Chart:** Timeline of tasks (requires special configuration).
- **Stock/Financial Chart:** Candlestick/stock price charts.
- **Pareto Chart:** A bar + line combo showing cumulative totals.

```
SELECT category AS label,
       SUM(value) AS series1,
       SUM(other_value) AS series2
FROM my_table
GROUP BY category;
```

All of these charts are demonstrated in the official Sample Charts app on apex.oracle.com. For example, the Bar chart sample page shows a “Stacked” and “Stacked Percent” example [8]. The Scatter page shows bubble sizes and zooming. Each chart type can be created by the Create Page wizard or in Page Designer by selecting “Chart” and then the specific type. Under the hood, APEX uses Oracle JET’s <oj-chart> component, so charts support animations, responsive resizing, tooltips, legends, and accessibility.
Chart configuration in APEX is partly declarative and partly code-driven. For instance, to create a bar chart, one writes a SQL Query such as: Then in the Chart region attributes (in Page Designer), map “label” to the X-axis and “series1”, “series2” to Y- axis series (color-coded). One can set Stack = Yes to stack these series together. If “Stack” is enabled, the chart adds values of series1 and series2 on top of each other for each category. The sample “Stacked Percent” chart uses a special configuration: by setting the primary axis Format to “Percent” and fixed 0–1 range, Oracle JET converts values to percentages of the group total. This shows that APEX automatically handles the normalization of stack values to percentages when requested.
The color palette of JET charts is integrated with Universal Theme. The bar chart sample notes that the default color palette can be changed with Theme Roller and will apply globally [8]. Moreover, one can override series colors declaratively: each series has a Color attribute, where a developer can pick from a color picker or bind to a SQL column (e.g., if the query returned a color alias) [8]. The sample also explains that colors can be further customized via JavaScript data filters if needed [8].

A novel approach for dashboards is to leverage Template Components (a feature introduced in 23.1 version) to create reusable chart containers or cards. For example, one could build a Template Component that defines a header, a placeholder, and an oj-chart instance via a plug-in attribute. Then, each dashboard page can simply drop that template component and supply the data source via its attributes. This encapsulates repetitive UI patterns and ensures consistency. The latest enhancements to Template Components (supporting unlimited attributes and faster rendering) [5] make this more practical. For instance, we could define a “Dashboard Tile” template that automatically applies padding, title style, and allows any chart region to occupy the tile content. Developers using the App Builder could then “Insert Template Component” for charts and skip manually adjusting margins and labels.

Example code snippet (SQL for a multi-series stacked bar):

Mapping: label→Category (X-axis), male_sal and female_sal as two bar series. With Stack = Yes, this would show each department's male and female salary stacked. We could add a Link column to this query, enabling drill-down (explained in next section).

Charts are not limited to database tables. Through APEX's Web Source Modules, a RESTful JSON feed can be defined as a new table source. For example, one could define a Web Source Module to fetch stock prices from a public API and then create a stock chart directly from that source. This highlights APEX's data integration: any REST or JSON data can feed into JET charts almost as easily as local SQL tables [4].
In summary, the charting capabilities of APEX are extremely rich. Developers can choose from virtually all common chart types and customize them with formatting, interactivity, and layout options (legends, dual axes, filler labels, etc.). Stackable series, percent displays, and automatic data densification for multi-series charts ensure smooth rendering [8]. The Oracle documentation and samples cover these features comprehensively [6]. In practice, dashboard designers will often combine charts of different types on one page – e.g. a bar chart next to a pie chart – to convey complementary views of the data.

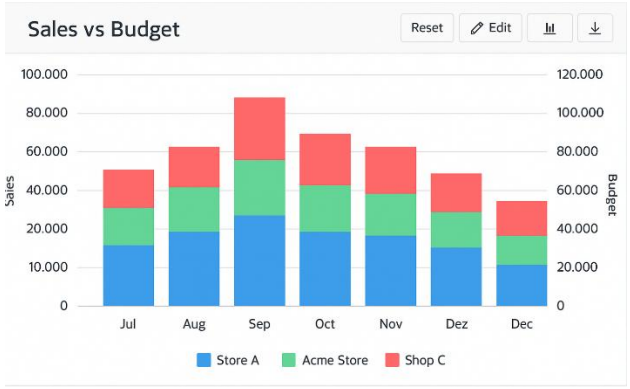

Figure 1. Stacked bar chart with dual axes.

IV. INTERACTIVITY AND DRILL-DOWN MECHANISMS

Beyond static charts, interactive dashboards require that users can click or filter to drill down into data. APEX supports several mechanisms for chart and report interactivity:

1. Chart Drill-Down via Links: As mentioned, each chart series or data point can have an associated link. In Page Designer, within a chart region's Series settings, there is a "Link" attribute. You can either choose to link to a page in the same app or specify a URL. The series-level link can include dynamic values using substitution strings or column aliases. For example, the built-in example shows linking to an Interactive Grid page. The URL behind the chart segment might look like f?p=&APP_ID.:29:&APP_SESSION.::NO:IG_DEPTNO:&DEPTNO., where &DEPTNO. is a column alias from the query [8]. The documentation describes exactly setting Target = Page 29 and Set Item IG_DEPTNO = &DEPTNO.. When a user clicks a bar, they are taken to the target page and the IG is automatically filtered to that department. This drill-through design leverages APEX's "URL Filtering" feature. The code for linking could also be placed in SQL, as:

```
SELECT category AS label,
       SUM(value) AS series1,
       SUM(other_value) AS series2
FROM my_table
         GROUP BY category;
```

and then in Chart Attributes set Link Column = LINK [8].

2. Faceted Search and Smart Filters: APEX brings advanced, declarative filtering into dashboards. A Faceted Search region can be set up on a table or view. It automatically creates sidebar facets for certain columns (e.g.

text search, list of values). Users can then click or type to narrow the data in a connected report. Input-type facets can use operators (like "not equals", "does not contain") thanks to the Arbitrary Facets feature [5]. For example, a text facet can be configured with a drop-down of operators, allowing end-users to easily invert filters. Smart Filters (from APEX 21.2) work like an enhanced search bar: users start typing and suggested filters appear. These components significantly reduce the need to code custom filtering logic. Figures 2 and 3 show example screenshots of these filters in action:

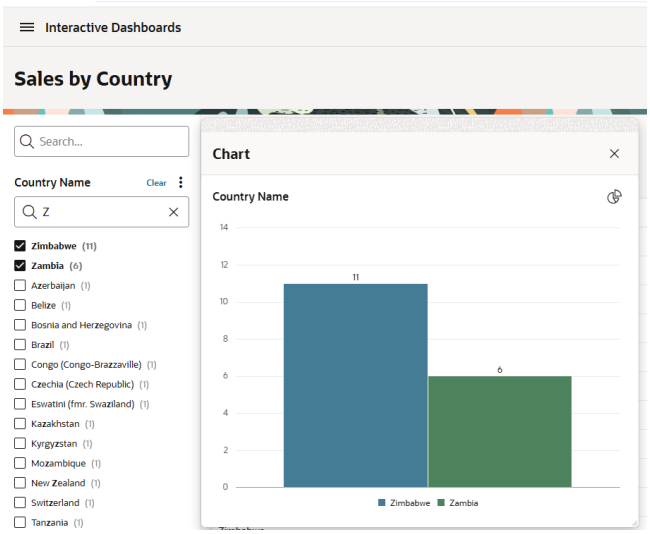

Figure 2. A faceted search region in APEX (left) with a "Country Name" filter.

When a filter is applied (e.g. selecting a facet value or entering text), all related components on the page (charts, reports) can automatically refresh if they reference the same underlying data. This creates a highly interactive dashboard without writing explicit DA code. In contrast, older approaches required manual re-fetching. Faceted and smart filters exemplify APEX's UI-driven interactivity.

3. Dynamic Actions and Custom JS: For ultimate flexibility, developers can use Dynamic Actions or custom JavaScript to make regions interactive. For example, as shown earlier, clicking a button fired a DA that used ojChart calls to re-orient a chart [9]. Similarly, one can use the JET API events: apex.region("myChart").widget().on("ojSelect", function(event, ui){ /* handle selection */ }); Custom JS can also be used to pass filters via apex.Util.setSessionState and refresh target regions. The combination of declarative links and programmable actions means any drill pattern can be implemented.

4. Cascading and Parameter Linking: Another common dashboard need is cascade filters. For example, selecting a region in a map might filter the following charts. APEX supports this via Page Items and Page Designer processes. One page item (hidden or visible) can hold a selection, and another chart's SQL can reference that item. When the first item changes (perhaps through a list selection or map marker click), the chart region can have a Refresh Dynamic Action. APEX's builder also allows Page Designer branching logic where setting a "Page Item to Submit" property on a chart region ensures that item's value is sent back on each refresh [14]. This automatic wiring means developers simply tie reports together by shared page items or use the automatic filtering in REST sources.

In the community and training materials, many interactive dashboard examples use APEX features above in combination, effectively demonstrating APEX's interactivity (if built today, it would use Faceted Search and charts with links). Another key UI pattern is carousel or tab containers to switch between charts while keeping filters persistent. This can be achieved by putting charts in a Tabs Container and linking their tab selection via DA to share page items.

In summary, APEX dashboards are not merely static charts, they are fully interactive analytic pages. The platform provides built-in drilldown linking, flexible filters, and event-driven programming. Figures 2 and 3 demonstrate the kind of UI/UX end-users experience. By combining these elements, complex analytics workflows (filter by X, then click chart for detail) are achievable with minimal coding. APEX's documentation

and samples showcase many interactive features [5], and developers can extend these with JavaScript for custom behavior.

V. UI/UX ENHANCEMENTS AND DATA INTEGRATION

A polished dashboard demands attention to UI/UX. APEX's Universal Theme and component framework greatly aid this. The theme's key principles are Responsive Design, Versatile UI components, and Easy Customization [3]. Universal Theme auto-adjusts layouts for mobile devices, and all charts and regions are responsive by default. For example, the Universal Theme includes a Tabs Container component – ideal for dashboard pages that organize multiple chart views in tabs. The Cards (media list) region can display key metrics or images at the top of the dashboard. APEX also offers an Icon List (horizontal bullet list with icons) for highlights, and Charts can have built-in headings and footers.

Theme Roller provides runtime theming. User can launch Theme Roller on a page to pick a global color palette that changes all charts and UI elements [8]. This means corporate branding can be applied across the app without editing CSS. The Template Options feature lets developers tailor component look without CSS: for instance, a report region has options to hide headings, add icons, or change pagination style. Several Template Components shipped with Universal Theme have added versatility [5]. Also, the enhancements to Template Directives allow advanced templating: text regions can now use {apply/} or {icon/} directives inline [5], enabling dynamic icon rendering or context switches in static HTML. For example, one can use {icon name = "success" class="text-green"} to embed an icon in a region's static text.

Accessibility is an important UX aspect. APEX automatically includes ARIA attributes for regions and chart tooltips, and the Redwood Light theme (updated in 23.2) improves color contrast [5]. Screen reader and keyboard navigation have also been refined. Developers should still test dashboards for accessibility, but APEX's modern frameworks handle much of the burden.

Screen layout is critical in dashboards. The Universal Theme provides grid classes (12-column grid, offsets, etc.) to arrange regions. For example, one can create a 3-column row of cards showing summary stats (each region width=4 columns), followed by a 2-column row where one side has a tall bar chart (width=8) and the other side two pie charts stacked (width=4). The Grid Layout region template simplifies this. The APEX builder's Page Designer also allows dropping Region Display Selectors so that multiple regions occupy the same position and user can toggle between them which is good for showing different charts in the same slot.

Another UI/UX advanced feature is Smart Filters, a search bar that located above reports/charts. They provide an alternative to faceted sidebars. For instance, an Outage Tracker dashboard might have Smart Filter for Outage Severity, Outage Status that persists across chart refreshes. Figures 3 shows a Smart Filter example:

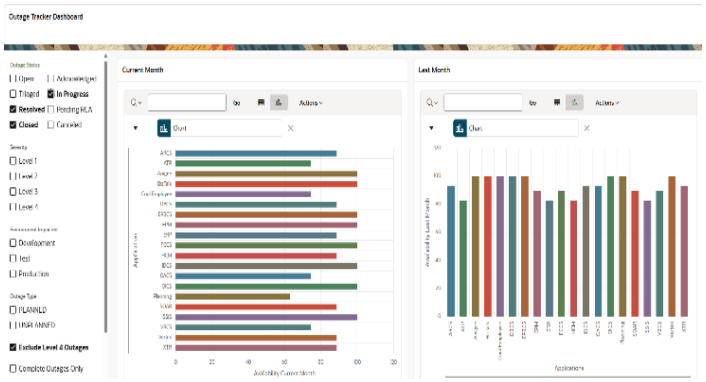

Figure 3. Example of a Smart Filter (top left) with search bar above the report/chart.

In addition to on-premises tables and rest data sources, APEX supports Web Source Modules as generic

REST/JSON data feeds. A Web Source can be defined from scratch with a URL; APEX will sample the JSON structure and create item lists. These can then be used as Tables in SQL queries. As [32] notes, any APEX component (Reports, Charts) can directly query a REST source. Moreover, APEX can synchronize remote data to local tables (via Scheduled Data Synchronization), enabling performant offline or large-volume reporting.

Oracle REST Data Services (ORDS): If an organization has an ORDS-enabled database, APEX can consume its endpoints natively, taking advantage of server-side filtering. For instance, a dashboard might use ORDS to fetch master-detail data in one call. - SQL-Enabled REST: APEX introduced "REST Enabled SQL". This lets a remote SQL query (on another database) be exposed via an ORDS service. APEX treats it as a REST Data Source with built-in pagination and filtering [4]. This is powerful for integrating with Autonomous Database or non-Oracle databases without ETL. Cloud SaaS: Support for Oracle Fusion/SaaS means one can directly visualize ERP or HCM data in APEX charts. The DSP blog [34] highlights how the new Oracle Cloud Applications SaaS REST Service type has built-in CRUD support, effectively letting dashboards include live transactional data from ERP/CRM systems [7]. - OData: APEX now has a native OData connector [5]. OData is common in Microsoft services and SharePoint. This means an APEX dashboard could query, for example, an Office 365 OData endpoint as easily as a local view. - External APIs: Any JSON or XML API can be consumed. For instance, a weather or geolocation API can enrich dashboard data with external metrics (e.g. plot sales by state on a map with current weather).

In addition to JSON/REST, APEX still supports some legacy sources: - CSV/Excel Data Load: Users can upload spreadsheets; the Data Load wizard can create a table and import data. Dashboards can then chart this table. - Database Links: One can query across databases using DBLink if configured. APEX charts treat it like any table query. - External Tables: If data is on a shared file system, APEX can use the DB's external table support.

All of these integration points mean dashboards can aggregate hybrid data (on-prem DB + cloud API + manual upload) with ease. The key is that once integrated, the data appears as just another table or view for the charts and filters.

To summarize UI/UX and integration: APEX focuses on making dashboard development as user- friendly as possible for both developer and end-user. For developers, the builder environment lets one drag- drop charts, set attributes, and preview instantly. The theme and template system reduce custom coding, and the new Smart Filters/facet features reduce the need to hand-code filtering. For end-users, the dashboards provide modern interactive controls (clickable chips, modals, mobile support). Data integration features ensure the underlying data can come from anywhere, making APEX a comprehensive dashboard solution. By leveraging these tools (as documented in Oracle's guides [3] [4] and community blogs [5] [7]), developers can craft sophisticated dashboards with exceptional UI/UX.

VI. FUTURE TRENDS AND EVOLUTION

While APEX is powerful, technology evolves rapidly. Looking ahead, we identify several trends and possible future features for APEX dashboards, especially focusing on UI/UX and data integration:

Advanced Data Visualizations: Users increasingly expect more sophisticated visuals (e.g. geospatial maps, network graphs, advanced analytics charts). Oracle APEX could further expand built-in chart types or integrate third-party libraries (like Vega or D3.js) more tightly. For example, adding a native Sankey diagram or chord diagram could help visualize flows or relationships. The APEX team might also consider a graph/network component for complex relationships. On the UI side, voice and gesture interfaces are emerging; perhaps future dashboards could support voice queries or AR overlays.

Real-Time and Streaming Data: Modern analytics often require near-real-time updates (stock tickers, IoT sensor data). APEX could introduce push channels (e.g. WebSockets or SignalR) to update dashboard charts live

without refresh. Oracle has DB notifications (AQ), but a declarative real-time chart update mechanism would be novel.
AI-driven Insights: With AI integration being hot, future APEX dashboards might embed machine learning insights directly. For example, automatic trend detection or predictive forecasts in charts, powered by Oracle Machine Learning or integrated Python libraries. APEX could offer a "Smart Chart Suggestion" where it analyzes data and suggests the best chart type or highlights anomalies. UI/UX might include natural language query boxes (ask the dashboard for a summary).

Dashboard Templates and Galleries: Improving the development experience with pre-built dashboard templates could help non-experts. APEX might include gallery pages with sample dashboard layouts, much like Power BI's templates. Coupled with Template Components, this would speed up building polished dashboards.
Mobile-First Dashboards: Although Universal Theme is responsive, dedicated mobile dashboard features (like a phone-friendly mobile layout editor) could enhance UX. For example, an iOS/Android app wrapper with offline sync, or native mobile touch gestures for charts.
Collaboration and Export: Future enhancements might include collaborative features (multiple users viewing/updating dashboards, commenting) and better exporting (interactive PDF or integration with tools like APEX Office Print). While one can currently print or export reports, making dashboards more shareable would be valuable.

These directions align with industry trends in BI and analytics. The overarching goal is that Oracle APEX will continue to evolve towards a comprehensive low-code analytics platform. The recent focus on UI components and data connectors suggests Oracle's commitment to make APEX dashboards even more powerful. We can reasonably expect future versions (e.g. 25.x) to incorporate some of these features, further solidifying APEX's role in the dashboard space.

VII. CONCLUSION

Oracle APEX offers a robust framework for building interactive, data-driven dashboards. By combining powerful charting with modern UI/UX components and versatile data integration, APEX enables developers to create analytics applications with minimal coding. The platform's drill-down and filtering capabilities (links in charts, interactive reports, faceted search) empower end-users to explore data intuitively. Code examples have been demonstrated for chart queries and dynamic actions (e.g., toggling chart orientation and stacking with JavaScript) [9], illustrating how easily APEX blends SQL and JS APIs.

On the UI/UX front, APEX's Universal Theme and template system allow rich, responsive designs without heavy CSS. Customizations like Theme Roller palettes and Template Directives let developers brand dashboards and fine-tune layouts [3]. New components like Smart Filters and enhanced facets (with custom operators) provide end-users with powerful, on-the-fly filtering [5].

Data integration remains a strong suit for APEX. The ability to consume RESTful endpoints, SQL queries over ORDS, and cloud SaaS sources means dashboards can incorporate any data. Official docs affirm that APEX charts and reports can connect directly to REST Data Sources [4]. This opens the door to combining on-premises and cloud data in one cohesive dashboard.

For database professionals and analysts, these features mean APEX dashboards can rival dedicated BI tools. Everything runs on Oracle's secure infrastructure, and the logic is written in familiar SQL and PL/SQL (or low-code configuration). Academic researchers can use APEX to build prototypes of data analytics platforms; students can learn data visualization on a platform that teaches both SQL and modern web principles.

In conclusion, Oracle APEX significantly advances the state of interactive dashboards on the Oracle stack. It combines a wide array of charts and controls with an intuitive builder interface. The platform's continual

enhancements to UI/UX (like responsive design and filters) and data connectivity (REST, JSON, SaaS) position it as a comprehensive solution for developing custom dashboards. Looking forward, APEX is poised to adopt emerging technologies (AI, real-time data, graph viz) which will further empower data professionals. For now, the combination of built-in capabilities and extensibility means that nearly any interactive dashboard requirement can be met within APEX.

ACKNOWLEDGMENT

The author thanks the Oracle APEX community for their extensive documentation, forums, and insightful blogs, which provided foundational insights for this research. The author would also like to disclose the use of the Grammarly (AI) tool solely for editing and grammar enhancements.

REFERENCES:

[1] M. Alhamadi, "Challenges, Strategies and Adaptations on Interactive Dashboards," in Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization, New York, NY, USA: ACM, Jul. 2020, pp. 368–371. Accessed: Jun. 28, 2025. [Online]. Available: https://doi.org/10.1145/3340631.3398678

[2] A. Angeioplastis, A. Tsimpiris, D. Varsamis, A. Baggia, and R. Leskovar, "Integration of ORACLE APEX Environment in Database Courses of Computer, Informatics and Telecommunications Engineering Department of International Hellenic University," in 42nd International Conference on Organizational Science Development, University of Maribor, University Press, 2023, pp. 13–23. Accessed: Jun. 28, 2025. [Online]. Available: https://doi.org/10.18690/um.fov.3.2023.2.

[3] Oracle, "Design Overview," Universal Theme. Accessed: Jul. 08, 2025. [Online]. Available: https://apex.oracle.com/pls/apex/apex_pm/r/ut/design-overview

[4] A. Chatterjee, "Understanding REST Data Sources," Oracle Help Center. Accessed: Jul. 08, 2025. [Online]. Available: https://docs.oracle.com/en/database/oracle/apex/23.1/htmdb/understanding-rest-data-sources.html

[5] Oracle, "What's New in APEX 23.2," Oracle APEX. Accessed: Jul. 08, 2025. [Online]. Available: https://apex.oracle.com/en/platform/features/whats-new-232/

[6] C. Cho, "App Builder User's Guide," Oracle Help Center. Accessed: Jul. 08, 2025. [Online]. Available: https://docs.oracle.com/en/database/oracle/application-express/20.1/htmdb/creating-charts.html

[7] B. Jackson, "Oracle APEX 23.2 New Features." Accessed: Jul. 08, 2025. [Online]. Available: https://content.dsp.co.uk/apex/oracle-apex-23.2-new-features

[8] Oracle, "Bar," Sample Charts (Built with APEX). Accessed: Jul. 08, 2025. [Online]. Available: https://apex.oracle.com/pls/apex/apex_pm/r/10800/bar

[9] N. T. Adjirackor, "Using Buttons to Show Horizontal, Vertical,Stacked and Unstacked Orientation of a Chart in an Oracle Apex Dashboard," Medium, Mar. 18, 2024. Accessed: Jul. 08, 2025. [Online]. Available: https://medium.com/@davidadjirackor/using-buttons-to-show-horizontal-vertical-stacked-and-unstacked-orientation-of-a-chart-in-an-6e9589562187

[10] Oracle, "How to Create a Drill Down Report," Oracle Database Application Express Documentation. Accessed: Jul. 08, 2025. [Online]. Available: https://docs.oracle.com/cd/E14442_01/doc/appdev.32/e13363/rprt_drill.htm

[11] P. K. Myakala, C. Bura, and R. Juma, *Interactive Data Dashboards: Design Principles, Best Practices, and Applications*. Preprint, 2024. [Online]. Available: https://www.researchgate.net/publication/388036825:contentReference[oaicite:109]{index=109}:contentReference[oaicite:110]{index=110}.

[12] B. Vogel, A. Kurti, M. Milrad, and A. Kerren, "An Interactive Web-Based Visualization Tool in Action: User Testing and Usability Aspects," in 2011 IEEE 11th International Conference on

Computer and Information Technology, IEEE, Aug. 2011, pp. 403–408. Accessed: Jun. 28, 2025. [Online]. Available: https://doi.org/10.1109/cit.2011.68

[13] Oracle, "Using Oracle Autonomous Database Serverless," Oracle Help Center. Accessed: Jul. 08, 2025. [Online]. Available: https://docs.oracle.com/fr-fr/iaas/autonomous-database-shared/doc/create-reports-analytics.html

[14] Oracle, "Creating Charts," Oracle Help Center. Accessed: Jul. 08, 2025. [Online]. Available: https://docs.oracle.com/database/apex-5.1/HTMDB/creating-charts.htm

[15] Oracle, "Dashboard," Sample Charts (Built with APEX). Accessed: Jul. 08, 2025. [Online]. Available: https://apex.oracle.com/pls/apex/r/apex_pm/sample-charts/dashboard