Programming for Business Analytics | Project 2

Event Planner Project

Student Name: Viraj Bharat Pahade

Student ID: 231049470

GitHub Repository Link: https://github.com/VirajPahade/EventPlannerProject

---

Introduction

Business and the Application

The Event Planner App aims to meet the operational needs of a small business that specializes in event planning and management. The nature of the business involves organizing and managing all types of events, such as weddings, corporate events, and parties, each of which should be handled smoothly with clients, vendors, and budgets.

The whole add-ins basically merge Microsoft Access, Excel, and VBA to keep key processes simple and automated. With the app, it ensures the integrity of data: that it's properly stored, efficiently retrieved, and effectively updated to support better decision-making.

Objectives of the Application

1. **Data Centralization**: Maintain a centralized database for managing clients, events, and vendors.
2. **Simple Event Tracking**: Provide a user-friendly interface in Excel for tracking event details, including dates, venues, budgets, and vendor assignments.
3. **Automation**: Automate repetitive tasks like importing data, running queries, and adding new events through VBA macros in excel.
4. **Data Analysis**: Enable business insights using pivot tables to analyze budgets, vendor contributions, and event types.

Database

The Event Planner App uses a Microsoft Access database to manage data related to clients, events, and vendors. The database consists of three interconnected tables and the detal information about the table is given below

1. **Clients Table**: Stores client details such as ClientID (Primary Key), Name, Contact, and Email.

2. **Events Table**: Tracks event details, including EventID (Primary Key), ClientID (Foreign Key), EventType, EventDate, Venue, Budget, and VendorID (Foreign Key).
3. **Vendors Table**: Manages vendor information, including VendorID (Primary Key), Name, Specialty, and Contact.

Relationship amongst databases

- **Clients ↔ Events**: One-to-many relationship linking clients to their events.
- **Vendors ↔ Events**: One-to-many relationship linking vendors to events.

Queries

In this project we perform 3 different queries to extract meaningful information.

1. **Events by Date**: Lists all upcoming events sorted by date.
2. **Top Vendors**: Identifies vendors servicing most events.
3. **Event Types**: Counts events by type (e.g., Weddings, Corporate).

The database allows smooth integration of data for effective event planning and important analysis.

Front – End

The front-end of the Event Planner App is built in **Microsoft Excel**, designed to provide a user-friendly interface for managing events, analyzing data, and interacting with the database. The Excel workbook consists of three primary tabs:

1. Events List Tab

This tab provides an organized view of all the events imported from the database. It includes key details such as EventID, Client Name, Event Type, Event Date, Venue, Budget, and Vendor Name. Conditional formatting is applied to highlight the status of events:

- Past events are shown in red.
- Upcoming events are shown in green.

By showcasing this data clearly, the Data Display Tab helps event planners keep track of schedules, prioritize tasks, and monitor progress immediately.

2. Analysis Tab

It features a pivot table that allows users to Analysis Tab is where event data is summarized and visualized.

- Analyze total budgets by event type.
- Review revenue contributions from different vendors.

- Filter data dynamically, such as by event date or vendor name, for deeper insights.

This tab gives planners a high-level view of their operations, helping them identify trends, allocate budgets effectively, and evaluate vendor performance.

3. Data Input Tab

The Data Input Tab acts as a form for adding new events into the system. It includes input fields for key details such as - Client Name, Event Type (with a dropdown list for consistency), Event Date, Venue, Budget, Vendor Name (also with a dropdown list).

There is also an **Add Event** button, which is linked to a VBA macro. When clicked, the macro validates the input and inserts the event details directly into the Access database. This tab ensures that event planners can quickly and accurately add new events without needing to interact with the database manually.

VBA Middleware

The VBA middleware is one of the main components of the Event Planner App that automates the communication between the Excel front end and the Access database. Middleware eliminates manual labor in retrieving, inserting, and analyzing data. A summary of some key subroutines used in the application is given below:

1. ConnectToDatabase

This subroutine establishes a connection between Excel and the Access database. Using the ADODB library, it connects to the database file and ensures that Excel can communicate effectively. A confirmation message is displayed to let the user know that the connection was successful.

- **Purpose**: To verify that the database connection is working and establish the link between Excel and Access.
- **Highlight**: The subroutine provides a simple way to test and ensure connectivity before performing any operations. Also, display confirmation message after successful connection.

2. ImportEvents

This subroutine is responsible for retrieving event data from the database and displaying it on the Events List Tab in Excel. It runs a query to fetch all event records, format the data, and organizes it neatly in the tab for easy viewing.

- **Purpose**: To keep the Events List Tab updated with the latest event information from the database.

- **Highlight**: Automates the process of importing data, saving time and reducing the risk of errors.

3. AddEventToDatabase

This subroutine takes input from the Add Event Tab and updates the database with the new event details. It validates the data entered by the user to ensure it is complete and accurate before inserting it into the database. If something goes wrong, it provides a meaningful error message.

- **Purpose**: To allow users to add new events directly from Excel without accessing the database manually.
- **Highlight**: Includes validation and error-handling features to maintain data integrity.

4. RefreshPivotTable

This subroutine refreshes the pivot table on the Analysis Tab after new data is added. It ensures that the analytics and insights reflect the latest event data.

- **Purpose**: To update the pivot table dynamically and ensure the summary reports are accurate.
- **Highlight**: Saves users the effort of manually refreshing the pivot table after every update.

Conclusion

The Event Planner App showcases how a small business can simplify event management by combining Microsoft Access, Excel, and VBA. By automating data handling and providing dynamic reporting tools, the app helps businesses save time, reduce errors, and make better decisions. It integrates essential features like event tracking, vendor management, and budgeting, making it a practical tool for day-to-day operations.

References

1. **Data Sources**: Sample data was self-created or adapted from publicly available templates.
2. **Lecture Slides**: Event Planner App concepts and technical details were supported by the lecture materials provided during the course.
3. **Tools Used**: Microsoft Access and Microsoft Excel (2016 or later).

AI Declaration

Portions of this project were completed with the assistance of AI tools (e.g., ChatGPT, Copilot, Gemini) to support brainstorming, structuring ideas, and refining content for the report. All the database design, VBA coding, and Excel implementation were completed by me, and the AI assistance was used as a learning and enhancement tool. I take full responsibility for the content of this project and have ensured that the final work meets the requirements and reflects my understanding.