AGENTIC AI IN POWER PLATFORM

Module 03

Power Apps Vibe – Copilot-First Creation

Prepared for EY Developers | BFSI Sector Delivery

Duration: 3 Hours (1 Hour Theory + 2 Hours Lab)

Status: Teachable (Preview) | Power Apps Vibe in Public Preview

CONFIDENTIAL | February 2026

1. Module Overview

This module introduces the Power Apps Vibe experience — Microsoft's next-generation app creation paradigm where makers describe what they want in natural language and receive fully functional, full-stack React-based applications. Vibe represents the "Copilot-first" creation model: instead of dragging and dropping controls onto a canvas, the maker collaborates with an AI copilot in a conversational loop to design, generate, and iteratively refine applications.

Important: Power Apps Vibe is currently in Public Preview (available at vibe.powerapps.com). It is not yet recommended for production use. All labs in this module should be framed as exploratory and prototype exercises. The UI and capabilities may change between course preparation and delivery.

Why This Module Matters

The Vibe experience changes how applications are born. In the traditional model, makers needed to understand controls, formulas, data sources, and layout before building. With Vibe, the maker's primary skill is the ability to articulate intent clearly. This is a fundamental shift that makes app development accessible to business analysts, subject matter experts, and citizen developers — the exact audience that BFSI organisations are trying to empower.

1.1 Learning Objectives

By the end of this module, participants will be able to:

1. Explain the Copilot-first creation paradigm and how it differs from traditional canvas and model-driven app development.
2. Navigate the Power Apps Vibe experience at vibe.powerapps.com and understand its interface components.
3. Generate a full-stack application from a natural language description using Vibe.
4. Understand the architecture of Vibe-generated apps (React-based front end, Dataverse back end).
5. Use conversational iteration to refine, modify, and extend a Vibe-generated application.
6. Describe how Vibe integrates with Plan Designer and Dataverse for holistic solution creation.
7. Identify the current capabilities, regional availability, and limitations of the Vibe experience.
8. Apply Vibe to rapidly prototype a BFSI-relevant application.

1.2 Module Structure

Section	Topic	Duration
Part A	Theory: Copilot-First Creation & Vibe Deep Dive	1 Hour
Part B	Hands-On Lab: Build a BFSI App with Vibe	2 Hours

1.3 Prerequisites

- Completion of Module 01 (The New Power Platform Vibe) and Module 02 (Plan Designer & Intent-Driven Architecture).
- Access to vibe.powerapps.com (the Vibe experience URL).
- Power Platform developer environment provisioned with a Dataverse database.
- Power Apps Premium license (trial is acceptable).
- Environment region set to US, Australia, Asia, or India (required for Vibe availability).
- Copilot features and preview features enabled in the tenant by an admin.
- Modern web browser (Microsoft Edge or Google Chrome, latest version).

2. Theory: Copilot-First Creation & the Vibe Experience

This section covers the conceptual foundations, architecture, and detailed capabilities of the Power Apps Vibe experience. Estimated duration: 1 hour.

2.1 The Evolution of App Creation in Power Platform

Power Apps has gone through three distinct generations of app creation experiences. Understanding this evolution is critical to appreciating the significance of the Vibe experience.

Generation	Experience	Key Characteristics
Gen 1 (2016–present)	Canvas Apps	Pixel-perfect control. Makers place controls on a canvas, write Power Fx formulas, and manage data connections manually. Full control but steep learning curve.
Gen 2 (2018–present)	Model-Driven Apps	Data-first approach. Apps are generated from Dataverse table metadata. Forms, views, and dashboards are configured, not coded. Structured but less flexible.
Gen 3 (2025–present)	Vibe Experience	Intent-first approach. Makers describe what they want in natural language. AI generates a full-stack React application with Dataverse back end. Iterative refinement through conversation.

Crucially, the Vibe experience does not replace canvas or model-driven apps. It adds a new creation paradigm on top of them. Canvas and model-driven apps remain the right choice for many scenarios — particularly when pixel-perfect control or deep customisation is required. Vibe is designed for rapid prototyping, new app creation, and scenarios where speed to first version matters more than fine-grained control.

> **Instructor Tip:** Ask participants which generation they have the most experience with. Most EY developers will be familiar with canvas apps and some model-driven apps. This helps set the context for why Vibe is a meaningful addition to their toolkit.

2.2 What Is the Power Apps Vibe Experience?

The Power Apps Vibe experience is a new app creation surface available at vibe.powerapps.com. It allows makers to describe a business application in natural language and generates a complete, functional application in response. The term “Vibe” reflects a broader industry trend of “vibe coding" — where developers describe the vibe or intent of what they want, and AI handles the implementation details.

2.2.1 Core Capabilities

Capability	Description
Natural Language App Creation	Describe your application in plain English. Vibe interprets the intent, designs the data model, generates the UI, and wires up the logic.
Full-Stack React Apps	Vibe generates modern React-based applications (not traditional canvas apps). These apps include responsive layouts, modern UI components, and client-side logic.
Automatic Dataverse Integration	Vibe automatically creates Dataverse tables (or connects to existing ones) to back the generated application. All CRUD operations are wired up.
Generative Pages	Each screen/page in the app is generated from AI. Pages include forms, lists, detail views, dashboards, and navigation — all generated from the natural language description.
Conversational Iteration	After the initial generation, makers can converse with the Vibe copilot to refine the app: "Add a status filter on the list page," "Change the colour scheme to blue," "Add a chart showing monthly trends."
Plan Integration	Vibe integrates with Plan Designer. A plan created in Plan Designer can be opened in Vibe for implementation, and Vibe can invoke Plan Designer to create data models on the fly.
Code Visibility	Makers can view and edit the generated React code directly if they want fine-grained control. This bridges the gap between no-code and pro-code.
Responsive Design	Generated apps are responsive by default, working across desktop, tablet, and mobile form factors.

2.2.2 How Vibe Differs from Traditional Canvas Apps

Dimension	Vibe Experience vs. Canvas Apps
Creation Method	Vibe: Natural language + AI generation. Canvas: Manual drag-and-drop with formulas.
Technology Stack	Vibe: React-based web application. Canvas: Power Apps proprietary runtime.

Formula Language	Vibe: JavaScript/React logic (generated by AI). Canvas: Power Fx formulas.
Data Layer	Vibe: Dataverse (auto-created). Canvas: Any data source (SharePoint, SQL, Excel, Dataverse, etc.).
Customisation	Vibe: Conversational refinement + code editing. Canvas: Studio editor with property panels.
Iteration Speed	Vibe: Very fast for initial creation and broad changes. Canvas: Faster for precise, pixel-level adjustments.
Production Readiness	Vibe: Preview — not yet recommended for production. Canvas: GA — enterprise-grade and production-ready.
Learning Curve	Vibe: Low (describe what you want). Canvas: Medium-High (learn controls, formulas, connectors).

Key Architecture Insight

Vibe-generated apps are not canvas apps. They are modern React-based web applications that run in the browser. This is a significant architectural departure. The generated code is real JavaScript/React code that makers (or pro developers) can inspect, understand, and modify. This makes Vibe-generated apps a bridge between the low-code world and the pro-code world.

2.3 The Vibe Interface

The Vibe experience interface at vibe.powerapps.com consists of several key areas:

Interface Area	Description
Prompt Area	The primary input area where makers type their natural language descriptions. This is where the conversation with the Vibe copilot happens.
Preview Pane	A live preview of the generated application. Updates in real time as the copilot generates or modifies the app.
Code Editor	A side panel that shows the generated React code. Makers can switch between visual preview and code view.
Data Panel	Shows the Dataverse tables and columns that back the application. Displays the data model created or connected by Vibe.

Plan View	When Vibe integrates with Plan Designer, this view shows the broader solution plan including tables, flows, agents, and other components.
Chat History	A conversation log of all prompts and responses between the maker and the Vibe copilot. Useful for understanding the evolution of the app.
Settings & Publishing	Controls for app settings, environment selection, and publishing the app to make it available to users.

2.4 The Vibe Creation Workflow

Creating an application with Vibe follows a natural, iterative workflow:

Step 1: Describe Your App

Enter a natural language description of the application you want. Be as specific as possible about the data, user roles, screens, and functionality.

Step 2: Review the Generated App

Vibe generates a complete application and displays it in the preview pane. Review the generated screens, data model, and functionality. Check that the layout, fields, and navigation match your expectations.

Step 3: Iterate Through Conversation

This is where the magic happens. You can converse with the Vibe copilot to refine the app. Each instruction modifies the existing app rather than regenerating from scratch. Examples of effective iteration prompts:

- “Add a date filter to the applications list page.”
- “Change the navigation to use a sidebar instead of top tabs.”
- “Add a summary card at the top of the dashboard showing total count, approved count, and pending count.”
- “Make the form fields for income and loan amount use currency formatting.”
- “Add form validation: email must be a valid format, loan amount must be greater than zero.”
- “Add a status badge next to each application — green for approved, red for rejected, yellow for pending.”

Step 4: Review the Data Model

Check the Data Panel to see what Dataverse tables and columns Vibe has created. Verify column types, relationships, and any auto-generated option sets. You can ask Vibe to modify the data model conversationally: “Add a ‘Branch’ column to the Applications table.”

Step 5: Optionally Edit Code

For pro-code scenarios, switch to the Code Editor to view or modify the generated React code directly. This is useful for adding custom business logic, integrating third-party libraries, or making changes that are easier to express in code than in natural language.

Step 6: Save and Publish

Once satisfied, save the app and publish it to make it available within the Power Platform environment. Published Vibe apps appear alongside canvas and model-driven apps in the Apps list.

> **Instructor Tip:** The iterative conversation is what makes Vibe powerful. Encourage participants to think of it as pair programming with an AI partner. The first generation is rarely perfect — but 3–5 rounds of conversational refinement can produce a surprisingly polished prototype.

2.5 Vibe and the Broader Power Platform Ecosystem

The Vibe experience does not exist in isolation. It is deeply integrated with the broader Power Platform and plays a specific role in the agentic development lifecycle:

Integration Point	How It Works
Plan Designer (Module 02)	Plans created in Plan Designer can be opened in Vibe for implementation. Vibe can also invoke Plan Designer to create Dataverse tables on the fly when the description implies a data model.
Dataverse	All Vibe-generated apps use Dataverse as their back end. Tables are auto-created or connected. This makes Vibe apps immediately compatible with the MCP architecture (Module 04).
Copilot Studio (Module 06)	Vibe-generated apps can be enhanced with embedded Copilot Studio agents for in-app AI assistance. The data model created by Vibe feeds into agent knowledge sources.

Agent Builder (Module 05)	While Agent Builder works with canvas apps, the patterns it teaches (extracting steps, rules, and triggers from app logic) apply conceptually to Vibe apps as well.
Power Automate	Vibe can generate apps that trigger Power Automate flows for backend automation (e.g., approval workflows, notifications).
MCP Dataverse (Module 04)	Tables created by Vibe are immediately accessible via the Dataverse MCP Server, enabling agents to perform CRUD operations on the same data the app uses.

2.6 Regional Availability and Licensing

Requirement	Details
Feature Status	Public Preview since 2025. Not yet recommended for production use.
Access URL	vibe.powerapps.com — this is the dedicated entry point for the Vibe experience.
Supported Regions	US, Australia, Asia, and India. The environment must be provisioned in one of these regions.
License	Power Apps Premium license (trial acceptable). A Dataverse database must be provisioned in the environment.
Copilot Settings	Copilot features and preview features must be enabled in Power Platform Admin Center.
Browser	Microsoft Edge or Google Chrome (latest version). No desktop installation required.

2.7 Current Capabilities and Limitations

As a Public Preview feature, participants should be aware of both the capabilities and the boundaries:

What Vibe Can Do Well

- Generate multi-screen CRUD applications with lists, forms, detail views, and navigation.
- Create modern, responsive UI layouts with cards, tables, charts, and status indicators.
- Auto-create Dataverse tables with appropriate columns and relationships.
- Handle iterative refinement through natural language conversation.
- Generate functional prototypes in minutes that would take hours or days to build manually.

- Provide code visibility for pro-dev customisation.

Current Limitations

- Not production-ready: Apps generated by Vibe should be treated as prototypes. They may lack error handling, security configuration, and edge case coverage.
- Complex business logic: Vibe handles straightforward CRUD and display logic well but may struggle with complex conditional logic, multi-step transactions, or intricate form dependencies.
- Offline support: Vibe-generated React apps require connectivity; they do not support offline-first patterns like traditional canvas apps.
- Advanced Dataverse features: Business rules, calculated fields, rollup fields, and security roles are not generated by Vibe.
- Connector support: Currently focused on Dataverse. Integration with external data sources (SharePoint, SQL, REST APIs) is limited compared to canvas apps.
- Accessibility: Generated apps may not fully meet WCAG accessibility standards without manual refinement.
- UI evolution: As a preview feature, the interface and capabilities are actively evolving. Screenshots and steps in this courseware may differ from the live experience.

> **Important:** Because Vibe is in Public Preview, the instructor should verify the live experience against this courseware 1–2 days before delivery. If significant UI changes have occurred, prepare a brief walkthrough addendum for participants.

2.8 BFSI Context: Vibe for Financial Services Prototyping

For BFSI organisations, the Vibe experience offers specific advantages in the prototyping and discovery phases:

- **During discovery workshops, describe a client’s process in real time and generate a working prototype on the spot. This is a powerful technique for stakeholder engagement.**Rapid Client Demonstrations:
- **Many BFSI processes (account opening, KYC, claims intake) are still paper-based or rely on Excel. Vibe can generate digital versions of these processes in minutes.**Process Digitisation:
- **When competing for BFSI projects, the ability to produce a working PoC in hours rather than weeks is a significant competitive advantage.**Proof of Concept Speed:

- **BFSI business analysts who understand the domain but lack coding skills can use Vibe to create functional prototypes that accurately reflect business requirements.**Business Analyst Empowerment:
- **Quickly prototype approval workflows, status tracking interfaces, and audit trail displays that BFSI compliance teams can review and provide feedback on.**Compliance Workflow Mockups:

Instructor Tip: For the BFSI audience, emphasize that Vibe is a prototyping and ideation tool, not a production deployment tool (yet). In BFSI, production apps require security hardening, audit trails, regulatory compliance, and thorough testing — all of which need to be added after the Vibe-generated prototype.

3. Module Summary

3.1 Key Takeaways

- The Power Apps Vibe experience represents the third generation of app creation: intent-first, Copilot-driven development.
- Vibe generates full-stack React-based applications from natural language descriptions, backed by auto-created Dataverse tables.
- The iterative conversation loop is the key differentiator — makers refine apps through dialogue, not manual configuration.
- Vibe integrates with Plan Designer for holistic solution creation and with Dataverse for MCP compatibility.
- Generated apps include real, inspectable React code, bridging the gap between no-code and pro-code development.
- Vibe is currently in Public Preview: excellent for prototyping and ideation, but not yet production-ready.
- For BFSI scenarios, Vibe is a powerful tool for rapid prototyping, client demonstrations, and process digitisation workshops.
- Canvas and model-driven apps remain essential for production scenarios requiring fine-grained control, offline support, or mature ALM.

3.2 Knowledge Check Questions

1. What is the Power Apps Vibe experience, and how does it differ from canvas app development?
2. What technology stack do Vibe-generated applications use?
3. List three types of iterative refinement prompts you can give to the Vibe copilot.
4. Why is Vibe not yet recommended for production use? Name two specific gaps.
5. How does Vibe integrate with Plan Designer (Module 02)?
6. What happens to the Dataverse tables created by Vibe? Can they be accessed via MCP?
7. In which regions is the Vibe experience currently available?
8. Describe one BFSI scenario where Vibe would accelerate solution delivery.
9. What is “vibe coding” and why is the code editor important even in a no-code tool?
10. When would you recommend canvas apps over Vibe for a BFSI client?

3.3 What’s Next

In Module 04: MCP Dataverse in Power Apps, you will learn how the Dataverse tables created by Plan Designer and Vibe become accessible to AI agents through the Model Context Protocol (MCP) Server. You will explore the built-in MCP tools (list_tables, describe_table, read_query, create_record, update_record, search_knowledge) and understand how agents can perform CRUD operations against Dataverse using natural language.

4. References and Resources

4.1 Priority 1 Sources (Primary)

- Microsoft Learn: "Overview of the New Power Apps Vibe Experience" — learn.microsoft.com/en-us/power-apps/vibe/overview
- Microsoft Learn: "Create Apps, Data, and Plans Together Using Power Apps Vibe" — learn.microsoft.com/en-us/power-apps/vibe/create-app-data-plan
- Microsoft Blog: "Inside the New Power Apps: The Future of App Development" (Ignite 2025) — microsoft.com/en-us/power-platform/blog/2025/11/18/inside-the-new-power-apps/

4.2 Priority 2 Sources (Supplementary)

- Microsoft Blog: "Introducing the New Power Apps: Generative Power Meets Enterprise-Grade Trust" — microsoft.com/.../introducing-the-new-power-apps-generative-power-meets-enterprise-grade-trust/

4.3 Priority 3 Sources (Additional)

- PowerApps911: "A New Way to Build Power Apps with Vibe Coding and AI" — powerapps911.com/post/a-new-way-to-build-power-apps-with-vibe-coding-and-ai

4.4 Related Course Modules

- Module 01: The New Power Platform Vibe (prerequisite, foundational concepts)
- Module 02: Plan Designer & Intent-Driven Architecture (prerequisite, Plan Designer integration)
- Module 04: MCP Dataverse in Power Apps (next module, Dataverse tables created by Vibe become MCP-accessible)
- Module 05: Agent Builder (converts app patterns into Copilot Studio agents)
- Module 06: Copilot Studio Agents (agents that work with Vibe-created data)