Project Time Estimation Assistant

To begin, I customized the system topic “Conversation Start” to better align with the goals of this assessment. I’ve also checked the “Fallback” answer for unrecognized inputs or errors, so the assistant can guide the user back on track.

I have also customized the assistant icon using an AI-generated image from *partyrock.aws*

Project Estimation Assistant

To structure the assistant, I first researched the following technologies and platforms. I selected three to build the assistant around:

- **Robotic Process Automation (RPA)** → *UiPath*
- **Low-code development** → *Appian*, a leading low-code automation platform known for its rapid application development capabilities.
- **Conversational AI** → *Cognigy*, which supports voice and chat automation and connects seamlessly with RPA and backend systems.

As a first step, once the “Welcome message” is displayed, I implemented a question in the assistant asking the user whether they would like a brief explanation of the available technologies and platforms before proceeding. This ensures clarity and helps users make informed choices, especially if they’re unsure. In the topic flow canvas, you’ll find this step called “Ask for explanation” question text box.

Once the user selected a Yes/No answer, and in case of Yes, the assistant shows a "Quick guide" with clear and concise explanations, the assistant clarifies that even though a project might involve more than one technology, the user should select the one most relevant as a starting point.

The current assistant estimates time based on a single selected platform, but I'm aware that real-world projects often combine multiple technologies (e.g., Conversational AI + RPA + low-code apps). In future iterations, the assistant could allow users to select multiple technologies and sum their base hours dynamically. This would involve adjusting the question flow and supporting multi-selection. While I chose to keep it simple for this version to avoid extra complexity, I noted this as potential future improvement.

For determining the time for each platform, I set:

- **UiPath**: 24 hours, considering the setup, building and developing of workflows, data handling, testing and debugging the results.
- **Appian**: 20 hours, as it's a rapid development platform with drag-and-drop components, reusable modules, and visual workflows. I have taken into consideration the workflow design and planning, building the screens, setting the workflow and testing.
- **Conversational AI**: 18 hours, based on estimates for dialogue and path design, intent/entity setup, testing and integration work.

I then defined the key criteria that impact project duration:

- Project complexity
- Number of integrated systems
- Level of data handling
- UI/UX requirements
- Security and compliance configuration

I added extra hours depending on the option chosen by the user:

Criteria	Level / Options	Description / Examples
Project Complexity	Simple	Straightforward workflows with few steps or branches. Example: Chatbot answering static FAQs.
	Medium (+10 hours)	Some branching, validations, 1–2 system integrations, user data input/output. Example: Power App submits requests and sends notifications.
	Complex (+20 hours)	Many conditions, loops, multiple integrations, complex logic. Example: Cognigy chatbot handling HR tasks + integrating with SAP.
Integrated Systems	1	Single system integration.
	2-3 (+10 hours)	Moderate number.
	More than 3 (+20 hours)	Multiple systems.
Data Handling	Low	No user data stored or processed. Example: static chatbot with predefined answers.
	Medium (+5 hours)	Simple data capture and routing, Example: form data saved to SharePoint or Excel.
	High (+15 hours)	Processing or transforming sensitive/unstructured data.
UI/UX Requirements	Basic UI	Minimal interface, standard forms or chat windows.
	Custom UI (+10 hours)	Tailored interface with branding, specific layouts, or custom controls. Example: a Power App with company branding and custom navigation.
	Multi-channel (+20 hours)	Solution accessible on multiple platforms (webchat, MS Teams, mobile). Example: Cognigy bot available via website chat and MS Teams.

Security & Compliance	None	No specific requirements. Default platform security is sufficient.
	Basic (+5 hours)	Needs access control setup.
	Advanced (+10 hours)	Requires encryption, authentication config.

While building the assistant, I encountered a limitation in Microsoft Copilot Studio regarding variable handling. Specifically, assigning different values (in hours) for each answer option couldn't be managed under a single dynamic variable. Instead, I had to create separate variables for each response, through a condition node, which adds complexity to the logic and maintenance of the assistant.

Additionally, although I configured estimated hour values for each response, Copilot Studio doesn't currently support performing arithmetic calculations across multiple variables. As a result, the values set for each answer aren't automatically reflected in the final result, and I had to simulate the estimated total using fixed conditions and static output.

Before displaying the final result, I added a step where the assistant appears to "calculate" ("Calculate Estimate" message), then displays a summary of the user's selections along with the estimated total time, including 20% buffer, in the "Final Output Message".

The current design allows for future improvements, such as handling multi-technology projects or integrating Power Automate for real-time calculations.

Please find below the link to the demo chat:

https://copilotstudio.microsoft.com/environments/Default-6f74127f-ca15-4c85-ba29-8a0a2e7e3cf3/bots/cra3d_projectEstimationAssistant/canvas?__version__=2&enableFileAttachment=true