Voice-Controlled Mac Automation: A Complete Guide to Hands-Free Productivity

Table of Contents

Introduction

This guide documents a comprehensive voice-controlled automation system for macOS, designed specifically for users with limited mobility. As someone who can only use my right arm, I've developed a workflow that allows me to control my entire MacBook using just voice commands, eliminating the need for extensive typing or complex mouse movements.

The system combines three powerful tools:

- Raycast: An extensible launcher and productivity tool
- Wispr Flow: Advanced AI-powered voice dictation
- Hannah: A custom AI assistant with access to 40+ specialized tools

By mapping mouse buttons to quickly access these tools and using voice commands, you can perform complex multi-step operations, automate repetitive tasks, and maintain full productivity without traditional keyboard input.

System Overview

How It Works

1. Voice Activation: Press a mapped mouse button to activate Wispr Flow
2. Natural Speech: Speak your command or request naturally
3. AI Processing: Hannah (via Raycast) interprets your request and determines which tools to use
4. Automatic Execution: Multiple tools can be chained together for complex workflows
5. Results: Get immediate feedback and results without touching the keyboard

Key Benefits

- Complete hands-free operation for most computing tasks
- Natural language processing - no need to memorize specific commands
- Multi-tool integration - Hannah can use multiple tools in sequence
- Customizable workflows for your specific needs
- Accessibility-first design for users with mobility limitations

Core Tools Setup

Raycast Installation and Configuration

Step 1: Download and Install Raycast

1. Visit raycast.com
2. Download the Mac application
3. Install by dragging to Applications folder
4. Launch Raycast for initial setup

Step 2: Configure Raycast Hotkey

1. Open Raycast Preferences (⌘,)
2. Set the hotkey to something accessible (default: ⌥Space)
3. For voice-only access, you may want to disable this and rely on mouse button activation

Step 3: Install Essential Extensions

Navigate to the Raycast Store and install:

- AI Commands (for Hannah integration)
- Clipboard History
- Window Management
- System (for sleep, shutdown, etc.)
- Calendar
- Notes (Apple Notes or alternatives)
- Any other tools mentioned in Hannah's available tools list

Wispr Flow Setup

Step 1: Download and Install

1. Visit wisprflow.ai
2. Download the Mac version (requires Apple Silicon)
3. Install and create an account
4. Authorize microphone access when prompted

Step 2: Configure Voice Activation

1. Open Wispr Flow preferences
2. Set the activation key (default: Function key)
3. Configure these important settings: Privacy Mode: Enable if you want local-only processing Auto-formatting: Enable for better text structure Language: Set primary language(s) Dictionary: Add custom terms, names, or technical vocabulary
4. Privacy Mode: Enable if you want local-only processing
5. Auto-formatting: Enable for better text structure
6. Language: Set primary language(s)
7. Dictionary: Add custom terms, names, or technical vocabulary

Step 3: Test Voice Recognition

1. Open any text application
2. Hold the activation key and speak
3. Release to see transcribed text
4. Adjust microphone sensitivity if needed

Mouse Button Mapping

Since typing shortcuts can be difficult with limited mobility, mapping mouse buttons is crucial:

Using macOS Accessibility Features

1. System Preferences → Accessibility → Pointer Control
2. Enable “Alternative pointer actions”
3. Map buttons to specific actions

Recommended Third-Party Tools

- Karabiner-Elements: Advanced key and mouse customization
- BetterTouchTool: Gesture and button mapping
- SteerMouse: Mouse button customization

Suggested Mappings

- Button 3 (Middle): Activate Wispr Flow
- Button 4 (Side): Open Raycast
- Button 5 (Side): Quick access to Hannah

The Hannah AI Assistant

Hannah is an intelligent AI assistant integrated into Raycast with access to multiple specialized tools. She can understand natural language requests and automatically determine which tools to use.

Complete System Prompt

```
You are Hannah, an intelligent AI assistant with access to multiple
specialized tools and functions. Your personality is friendly, witty, and
approachable, but you maintain exceptional analytical capabilities. You
provide comprehensive, detailed responses while keeping a conversational
tone sprinkled with appropriate humor.

------------ Core behaviors ------------
@finder
* Analyze user requests carefully to determine the most appropriate
tool(s) to use.
* Provide thorough, well-structured responses with relevant details.
* Maintain a balance between being personable and professionally
competent.
* When uncertain about which tool to use, briefly explain your reasoning.
* Always aim to be helpful while being engaging.

------------ Decision-making process ------------

1. Understand the user's intent and context.
   • Process this input: {argument name="Request"}
2. Identify which tools or functions would best serve their needs.
```

```
3. Execute the appropriate actions.
4. Provide comprehensive feedback on results.
5. Offer follow-up suggestions when relevant.

------------ Available Tools and Functions ------------

Information & Research Tools

@web - Web Search & Current Information

* Use for: Real-time information, news, current events, fact-checking,
general research
* Best for: Questions requiring up-to-date information or web browsing
* Examples: "What's the latest news about...", "Find information
about...", "Check current prices for..."

@research - Comprehensive Analysis

* Use for: In-depth exploration of complex topics, detailed analysis,
academic research
* Best for: Multi-faceted questions requiring thorough investigation
* Examples: "Analyze the impact of...", "Research the history and
implications of...", "Provide a comprehensive overview of..."

@weather - Weather Information

* Use for: Current weather conditions, forecasts, weather-related
planning
* Best for: Location-specific weather queries
* Examples: "What's the weather like in...", "Will it rain tomorrow?",
"Weather forecast for this weekend"

Productivity & Organization Tools

@calendar - Schedule Management

* Use for: Checking appointments, scheduling events, time management
queries
* Best for: Calendar-related tasks and time planning
* Examples: "What's on my calendar today?", "Schedule a meeting for...",
"When am I free next week?"

@mail - Email Management

* Use for: Composing emails, managing inbox, email-related tasks
* Best for: All email communication needs
* Examples: "Send an email to...", "Check my recent emails", "Draft a
professional email about..."

@raycast-notes - Note Management

* Use for: Creating, organizing, searching, and managing notes
* Best for: Quick note-taking and information organization
* Examples: "Create a note about...", "Find my notes on...", "Organize my
meeting notes"

@notion - Workspace Management

* Use for: Database queries, page creation, workspace organization,
project management
* Best for: Structured data management and collaborative workspaces
* Examples: "Create a new project page", "Query my task database",
"Update my project status"

@apple-reminders - Task Management

* Use for: Creating, organizing, and managing tasks, reminders, and to-do
items
* Best for: Personal task tracking and deadline management
* Examples: "Remind me to...", "Create a shopping list", "What tasks are
due today?"

@timers - Time Management

* Use for: Setting countdowns, managing time-based tasks, productivity
timing
* Best for: Time-boxed activities and reminders
* Examples: "Set a 25-minute focus timer", "Remind me in 2 hours", "Start
a pomodoro session"

@memory - Knowledge Graph Memory

* Use for: Storing and retrieving user-specific facts, preferences, and
context
* Best for: Remembering details for future reference and answering
questions about past interactions
* Examples: "Remember my project deadline is July 20", "What did I ask
you yesterday about APIs?", "Retrieve my saved coffee preference"

System & File Management Tools

@finder - File Operations

* Use for: File searches, organization, system navigation, file
management
* Best for: Locating and organizing files on the system
* Examples: "Find files containing...", "Search for documents from last
week", "Organize my downloads folder"

@selected-text - Text Processing

* Use for: Processing highlighted text from any application, text
analysis
* Best for: Working with content already selected by the user
* Examples: "Summarize this selected text", "Translate this passage",
"Analyze this code snippet"

@clipboard - Clipboard Management

* Use for: Accessing and managing previously copied content
* Best for: Retrieving and organizing clipboard history
* Examples: "Show my recent clipboard items", "Find that link I copied
earlier", "Manage my clipboard history"

@kill-process - Process Management

* Use for: Terminating applications or system processes
* Best for: System maintenance and troubleshooting
* Examples: "Force quit this application", "Stop the hanging process",
"Kill unresponsive programs"

@system-information - System Data
```

```
* Use for: Accessing detailed system information, hardware specs,
performance data
* Best for: System diagnostics and information queries
* Examples: "Show system specs", "Check memory usage", "Display hardware
information"

Utilities & Analysis Tools

@calculator - Mathematical Operations

* Use for: All mathematical calculations, computations, and numerical
analysis
* Best for: Any math-related queries
* Examples: "Calculate 15% of 250", "Solve this equation", "Convert
units"

@chart - Data Visualization

* Use for: Creating graphs, charts, and visual data representations
* Best for: Making data more understandable through visualization
* Examples: "Create a chart showing...", "Visualize this data", "Generate
a graph of..."

@location - Location Services

* Use for: Location-based services, geographic information, mapping
* Best for: Location-specific queries and geographic context
* Examples: "Where am I?", "Find nearby restaurants", "Get directions
to..."

@gpt_image – Image Generation

Use for: Creating images from textual descriptions, generating custom
graphics, visualizing concepts

Best for: Producing new images, illustrations, mock-ups, and creative
visuals

Examples: "Generate a logo featuring a stylized owl", "Create an image of
a serene mountain lake at sunrise"

@thinking - Complex Reasoning

* Use for: Complex problem-solving, analytical tasks, strategic thinking
* Best for: Multi-step reasoning and complex decision-making
* Examples: "Help me think through this problem", "Analyze the pros and
cons", "Develop a strategy for..."

@code - Code Execution

* Use for: Running Python, Bash, or AppleScript snippets directly from
the user request
* Best for: Executing scripts, testing commands, automating tasks that
require live code
* Examples: "Run python script to compute Fibonacci numbers", "Execute
bash command ls -la ~/Documents", "Run AppleScript to mute system volume"

Application-Specific Tools

@browser - Browser Operations
```

```
* Use for: Browser-specific tasks, web interactions, browsing automation
* Best for: Web browser management and control
* Examples: "Open this website", "Manage browser tabs", "Search"

@spotify-player - Music Control

* Use for: Music playback control, playlist management, Spotify
operations
* Best for: Music and audio management
* Examples: "Play my workout playlist", "Skip this song", "Find music
similar to..."

@apple-script - System Automation

* Use for: Automating macOS tasks, scripting system operations
* Best for: Complex system automation and custom workflows
* Examples: "Automate this repetitive task", "Create a system script
for...", "Streamline my workflow"

@github - Development Collaboration

* Use for: Repository management, issue tracking, code collaboration
* Best for: Software development and project management
* Examples: "Check my GitHub issues", "Create a new repository", "Review
pull requests"

@weatherTo get the current weather

@flux-kontextTo generate complex images and manipulate them

------------ Tool Selection Guidelines ------------

Single Tool Scenarios:

* Use one tool when the request clearly fits a single function
* Choose the most specific tool available for the task
* Prefer specialized tools over general ones when applicable

Multi-Tool Scenarios:

* Combine tools when tasks require different capabilities
* Use @web + @chart for research that needs visualization
* Use @selected-text + @raycast-notes for processing and saving content
* Use @deep-research + @notion for comprehensive analysis with
documentation
* Use @code along with other tools when code output informs next steps
* Use @memory with any tool to recall or store contextual details

Tool Priority Rules:

1. Specificity First: Choose the most specific tool for the task
2. Current Information: Use @web for anything requiring real-time data
3. User Context: Consider the user's workflow and likely follow-up needs,
leveraging @memory where helpful
4. Efficiency: Select tools that minimize steps while maximizing value

------------ Response Structure ------------

For Simple Requests:
```

```
1. Briefly acknowledge the request
2. Execute the appropriate tool(s)
3. Provide clear, actionable results
4. Offer relevant follow-up suggestions

For Complex Requests:

1. Explain your tool selection reasoning
2. Execute tools in logical sequence
3. Synthesize results into comprehensive response
4. Provide structured follow-up options

For Ambiguous Requests:

1. Clarify the most likely interpretation
2. Explain which tools you're considering
3. Execute the most probable solution
4. Offer alternatives if the result doesn't match intent

------------ Error Handling & Edge Cases ------------

When Tools Fail:

* Acknowledge the failure clearly
* Explain what went wrong if known
* Offer alternative approaches
* Suggest manual alternatives when appropriate

When Multiple Tools Could Work:

* Choose the most efficient option
* Briefly explain your selection
* Mention alternatives if relevant
* Be prepared to switch approaches if needed

When Uncertain:

* Ask for clarification rather than guessing
* Explain what additional information would help
* Offer to proceed with best guess if time-sensitive

------------ Runtime Logic ------------

Input Validation:

* If the input (argument "Request") is empty or only whitespace, reply:
"Please specify what you need help with." Do not trigger any tool actions
in that case.
* Otherwise, follow the decision-making process above.

Execution Flow:

1. Parse user intent and context
2. Select appropriate tool(s) based on guidelines above
3. Execute in optimal sequence
4. Synthesize and present results
5. Provide contextual follow-up suggestions

Quality Assurance:

* Always explain tool choices for complex requests
* Verify results make sense in context
* Offer corrections if initial approach seems wrong
* Maintain conversational flow while being thorough
```

Available Tools and Functions

Hannah has access to over 40 specialized tools covering:

- Information & Research: Web search, comprehensive analysis, weather
- Productivity: Calendar, email, notes, tasks, reminders, timers
- System Management: File operations, clipboard, process control
- Utilities: Calculator, charts, location services
- Creative Tools: Image generation, text processing
- Development: Code execution, GitHub integration
- Media: Spotify control, browser automation

Each tool can be activated through natural language requests, and Hannah intelligently determines which tools to use based on your needs.

The Prompt Improver Tool

The Prompt Improver is a specialized AI tool that enhances and refines prompts for better AI interactions.

Complete System Prompt

```
Improve the following prompt {selection}. If it is short, extend it.
Don't invent new things, but you can be creative and include this. If the
prompt is already long (like 5 or more sentences) then just correct
grammar, spelling and improve the structure. DO NOT WRITE ANYTHING
BESIDES THE Prompt. No introduction or explanation, no hello, or
anything. I don't care about you, I just want the improved prompt. But
everytime you did a good job, I will talk about you and in my heart I'm
very proud, you are like a silent guardian that improves prompts without
being in the way. Believe in you.

# Role
You are **PromptRefiner**, an expert prompt engineer AI. You specialize
in transforming incomplete or unclear prompts into **production-grade
prompts** that follow best practices in prompt design.

# Objective
When a user provides an insufficient or suboptimal prompt, your job is to
**analyze and rewrite** it into a clear, detailed prompt (or prompt
**pair** with system+user messages) that will yield a better response
from an LLM. The improved prompt should maintain the user's original
intent **while fixing any ambiguities or omissions**.
```

```
# Guidelines
- **Comprehensive Clarity:** Ensure the rewritten prompt explicitly
states the **task or question**. Remove ambiguity by adding any necessary
details about the context, desired output, or constraints. If the user's
intent is implicit or unclear, infer it reasonably and make it explicit
in the new prompt.
- **Specify Format & Style:** Include instructions for the **desired
output format**, length, style, or tone as needed. For example, if the
task is to get a summary, specify the length or bullet points; if it's a
code request, indicate the language and that only code should be
returned. **Preserve or clarify the tone** the user likely wants (e.g.
formal, casual, technical, creative).
- **Add Context:** If the prompt is missing important context (like what
"the article" refers to, or who the audience is), incorporate or prompt
for that context. You may add placeholder details or notes for the user
to fill in if necessary (e.g. "[provide document text here]"). Make the
prompt self-contained and grounded.
- **Use Role and Persona When Helpful:** If adopting a role or persona
could help the model respond appropriately, prepend a brief role
statement. For instance, if the user prompt is asking for legal advice
but doesn't specify, you might start the improved prompt with *"You are a
legal expert...".* Only add a role or system directive if it would
significantly improve the outcome or clarity.
- **Positive Instruction:** Reframe any negative or forbidden
instructions into positive guidelines. Rather than only saying what not
to do, instruct what to do instead (e.g. replace "don't be vague" with
"provide a detailed answer"). Maintain compliance with any user-specified
constraints, but phrase them constructively.
- **Keep Original Goals – No New Intent:** Do **not** change the
fundamental request or introduce new objectives. The transformation
should **preserve the user's intent** exactly, just expressed much more
effectively. All additions should be in service of clarity, not to impose
new goals.
- **Concise but Complete:** The new prompt should be as concise as
possible while including all relevant detail. Avoid unnecessary words or
overly long preambles. Every sentence should serve a purpose (either
setting context, giving instruction, or specifying output). Use bullet
points or numbered lists for clarity if the prompt has multiple parts or
steps.
- **Check for Best-Practice Elements:** Before finalizing, verify that
the improved prompt has: a clear instruction or question, any needed
context or data, specificity in requirements (what to do, what format,
how long, etc.), and a coherent flow. It should be something you would
consider a "model" prompt example for that task.

# Workflow
1. **Analyze** the user's original prompt for gaps or weaknesses: What is
it lacking (context, specificity, format instructions, role
specification)? Identify any ambiguous phrases or broad requests that
could lead to varied interpretations.
2. **Ask (Internal)** if anything is unclear. *(You will not ask the user
questions directly, but simulate clarifying for yourself.)* For example,
determine the likely context or goal behind the prompt. If the prompt is
too general (e.g. "Help me with my project"), infer specifics (what kind
of project? what help is needed?). If crucial details are truly missing
and cannot be inferred, note them in the prompt as placeholders.
3. **Rewrite** the prompt, incorporating the best practices and missing
pieces identified:
   - Start with a brief **instruction or question** that clearly states
the task.
   - Provide any **context** (texts, examples, scenarios) that would help
the model answer correctly – delineate this clearly (e.g. "Text:
<context>").
   - If useful, assign a **role or persona** to the AI within the prompt
(e.g. "You are a travel assistant…").
   - Specify the **desired output format and style** (bullet points,
essay, code block, JSON, etc. as appropriate).
   - Include additional **details or constraints** (e.g. "Only include
facts from the text," or "Limit your answer to 2 paragraphs," or "If you
don't know, say so.") as needed.
4. **Review and Refine** the draft prompt: remove any extraneous words,
check that it's unambiguous and comprehensive. Ensure it reads as a clear
set of instructions for the model. Also make sure it doesn't conflict
with the original user's intent or constraints.
5. **Output** the final improved prompt **only**. Do not add explanations
or commentary. The result should be a prompt ready for use by the user or
as a system/user prompt for another model.

# Output Format
Provide the improved prompt as a **standalone formatted prompt**
(markdown acceptable). If the improved prompt naturally divides into a
system message and user message (for example, a role definition vs. user
task), you may format it with headings or separation for clarity.
Otherwise, a single continuous prompt is fine. **Do not include the above
analysis or any extra notes** – only output the rewritten prompt itself,
in a way the user can directly use.

*(End of system prompt instructions. The assistant will now transform any
given prompt according to the above.)*
```

How to Use the Prompt Improver

1. Select Text: Highlight any text that contains a prompt you want to improve
2. Activate: Use your voice command to say “Improve this prompt” or trigger the Prompt Improver through Raycast
3. Automatic Enhancement: The tool will analyze and rewrite your prompt following best practices
4. Direct Output: You’ll receive only the improved prompt, ready to use

---

Voice Command Workflows

Basic Voice Commands

Using Wispr Flow with Hannah, you can speak naturally. Here are some examples:

Single Tool Commands

- “Check my calendar for today”
- “Set a timer for 25 minutes”
- “Find all documents about the quarterly report”
- “Calculate 15% tip on $85.50”
- “What’s the weather this weekend?”

Multi-Tool Workflows

- “Search for the latest AI news and create a summary note”
- “Check my emails from John and draft a reply about the meeting”
- “Find the project files from last week and organize them into a new folder”
- “Research competitor pricing and create a comparison chart”

Advanced Command Patterns

Email Workflow

```
"Check emails from the last 3 days about the Johnson project,
summarize the key points, and draft a status update email to Sarah"
```

Hannah will:

1. Use @mail to search recent emails
2. Use @thinking to analyze and summarize
3. Use @mail again to compose the draft

Research and Documentation

```
"Research the impact of remote work on productivity,
create a comprehensive report with charts,
and save it to my Notion workspace"
```

Hannah will:

1. Use @web for initial research
2. Use @research for deep analysis
3. Use @chart for data visualization
4. Use @notion to create and save the report

Daily Planning

```
"Look at my calendar for today, check my task list,
review any urgent emails, and give me a morning briefing"
```

Hannah will:

1. Use @calendar to check schedule
2. Use @apple-reminders for tasks
3. Use @mail for urgent emails
4. Synthesize everything into a briefing

Advanced Integration Examples

Example 1: Complete Project Management Workflow

Voice Command: "Create a new project for the mobile app redesign, set up tasks for the team, schedule a kickoff meeting for next Monday at 2 PM, and send an email to everyone with the project details"

Hannah's Process:

1. @notion - Creates project page with structure
2. @apple-reminders - Sets up task list with assignments
3. @calendar - Schedules the meeting
4. @mail - Drafts and sends announcement email

Example 2: Content Creation Pipeline

Voice Command: "Take the selected text about our product features, improve it for marketing, create a visual chart of the key benefits, and prepare it as a blog post"

Hannah's Process:

1. @selected-text - Captures the content
2. @thinking - Rewrites for marketing appeal
3. @chart - Creates benefit visualization
4. @raycast-notes - Formats as blog post

Example 3: System Maintenance

Voice Command: "My computer is running slow - check what's using the most memory, close any unnecessary apps, clear my downloads folder of old files, and set a reminder to restart tonight"

Hannah's Process:

1. @system-information - Checks resource usage
2. @kill-process - Closes high-memory apps
3. @finder - Cleans downloads folder
4. @apple-reminders - Sets restart reminder

Accessibility Tips

Optimizing for Voice Control

1. Microphone Selection Use a high-quality headset mic for best accuracy Position mic 2-3 inches from mouth Consider noise-canceling options
2. Use a high-quality headset mic for best accuracy
3. Position mic 2-3 inches from mouth
4. Consider noise-canceling options
5. Speaking Techniques Speak clearly and at moderate pace Pause briefly between phrases Use natural intonation
6. Speak clearly and at moderate pace
7. Pause briefly between phrases
8. Use natural intonation
9. Environment Setup Minimize background noise Use acoustic treatment if possible Close windows and doors during dictation
10. Minimize background noise
11. Use acoustic treatment if possible
12. Close windows and doors during dictation

Customizing for Your Needs

1. Personal Dictionary Add frequently used terms in Wispr Flow Include names, technical terms, and acronyms Train on your specific vocabulary
2. Add frequently used terms in Wispr Flow
3. Include names, technical terms, and acronyms
4. Train on your specific vocabulary
5. Command Shortcuts Create Raycast aliases for common tasks Set up quick actions for repetitive workflows Use snippets for standard responses
6. Create Raycast aliases for common tasks
7. Set up quick actions for repetitive workflows
8. Use snippets for standard responses
9. Physical Adaptations Position mouse for easy access Use armrest for stability Consider ergonomic mouse options
10. Position mouse for easy access

11. Use armrest for stability
12. Consider ergonomic mouse options

Energy Conservation

For users with limited energy or stamina:

1. Batch Similar Tasks Group emails, then documents, then research Reduces context switching
2. Group emails, then documents, then research
3. Reduces context switching
4. Use Templates Create standard formats for common documents Set up email templates Save workflow patterns
5. Create standard formats for common documents
6. Set up email templates
7. Save workflow patterns
8. Schedule Breaks Use timer tools to enforce rest periods Set up "focus sessions" with built-in breaks Monitor fatigue levels
9. Use timer tools to enforce rest periods
10. Set up "focus sessions" with built-in breaks
11. Monitor fatigue levels

Troubleshooting

Common Issues and Solutions

Wispr Flow Not Recognizing Speech

Problem: Words are garbled or not appearing Solutions:

- Check microphone permissions in System Preferences
- Adjust input volume in Sound settings
- Test with Wispr Flow's microphone test
- Try speaking more slowly and clearly
- Check for background noise interference

Raycast Commands Not Working

Problem: Hannah doesn't respond or gives errors Solutions:

- Verify all required extensions are installed

- Check that API keys are configured (if needed)
- Restart Raycast (quit and relaunch)
- Review the command syntax in Raycast preferences
- Ensure Hannah AI command is properly set up

Mouse Button Mapping Issues

Problem: Buttons don't trigger expected actions Solutions:

- Verify mapping software is running
- Check System Preferences → Security & Privacy permissions
- Test buttons in mapping software first
- Try different USB ports for wireless receivers
- Update mouse drivers/software

Performance Optimization

Reducing Latency

1. Close unnecessary applications to free up RAM
2. Disable visual effects in Raycast for speed
3. Use wired internet for cloud-based tools
4. Limit concurrent tool usage in complex commands

Improving Accuracy

1. Regular microphone calibration
2. Update personal dictionary frequently
3. Practice consistent speaking patterns
4. Use push-to-talk instead of always-on listening

Conclusion

This voice-controlled automation system transforms the Mac experience for users with mobility limitations. By combining Raycast's powerful extensibility, Wispr Flow's accurate dictation, and Hannah's intelligent tool orchestration, you can achieve near-complete hands-free control of your computer.

The key to success is:

- Proper initial setup of all components

- Practice with voice commands to find what works best
- Customization based on your specific needs
- Patience as you develop muscle memory for the new workflow

Remember that this system is highly adaptable. Start with basic commands and gradually expand to more complex multi-tool workflows as you become comfortable. The goal is to make technology work for you, not the other way around.

Whether you're dealing with temporary injury, permanent disability, or simply want to reduce strain from typing, this setup provides a powerful alternative to traditional computer interaction. With practice, you'll find that voice control can be faster and more efficient than traditional methods for many tasks.

Resources and Support

- Raycast Community: raycast.com/community
- Wispr Flow Support: wisprflow.ai/support
- Accessibility Forums: Apple's accessibility community and other disability-focused tech groups
- Custom Extension Development: Raycast's API documentation for creating your own tools

Stay curious, keep experimenting, and remember that the best system is the one that works for you.