# Cursor AI Prompt for Jarvis Desktop Assistant

Create a complete Electron desktop assistant application called "Jarvis" with the following specifications:

## Core Features Required:

1. \*\*Porcupine Wake Word Detection\*\* - Use @picovoice/porcupine-node for "Hey Jarvis" activation

2. \*\*Voice Recognition\*\* - Web Speech API for command recognition after wake word

3. \*\*System Control\*\* - Execute system commands (open apps, control windows, etc.)

4. \*\*Modern UI\*\* - Frameless window with arc reactor animation

5. \*\*System Tray\*\* - Minimize to tray with context menu

## Project Structure:

```

jarvis-desktop/

├── package.json

├── .env.example

├── README.md

├── src/

│ ├── main/

│ │ ├── index.js # Main process

│ │ ├── preload.js # Preload script

│ │ ├── porcupine-manager.js # Wake word detection

│ │ └── system-commands.js # Command execution

│ └── renderer/

│ ├── index.html # UI

│ ├── styles.css # Styling

│ └── renderer.js # Renderer process

└── assets/

├── icon.png

├── icon.ico

└── tray-icon.png

```

## Technical Requirements:

- Electron 27+

- Node.js 18+

- Porcupine wake word detection (requires API key)

- Cross-platform (Windows, macOS, Linux)

- No external AI APIs needed

- All processing done locally

## Key Implementation Details:

### 1. Porcupine Integration:

- Initialize with access key from .env

- Use built-in "JARVIS" keyword

- Continuous background listening

- Emit event on wake word detection

### 2. Voice Commands to Support:

- "Open [app name]" (Chrome, Firefox, Notepad, Calculator)

- "What time is it?"

- "What's the date?"

- "Take a screenshot"

- "Lock the computer"

- "Search for [query]"

- "System info"

- "Create new file"

### 3. UI Features:

- Frameless transparent window

- Arc reactor animation that pulses on activation

- Chat-style message display

- Manual text input option

- Minimize/close buttons

- Always-on-top option

### 4. System Tray Features:

- Show/hide window

- Start/stop listening

- Settings access

- Quit application

### 5. Settings (.env):

```

PORCUPINE\_ACCESS\_KEY=your\_key\_here

WAKE\_WORD=jarvis

SENSITIVITY=0.5

ALWAYS\_LISTENING=true

MINIMIZE\_TO\_TRAY=true

```

## Build Configuration:

- Use electron-builder for packaging

- Include app icons for all platforms

- Sign and notarize for macOS

- Auto-updater support

## Error Handling:

- Graceful fallback if Porcupine fails

- Microphone permission checks

- Command execution error handling

- User-friendly error messages

## Performance Considerations:

- Efficient audio processing

- Minimal CPU usage when idle

- Quick wake word response time

- Smooth UI animations

Please generate the complete application with all files, focusing on reliability and user experience. The app should work immediately after npm install (except for adding Porcupine API key).