**Citizen AI – CUSTOMER JOURNEY MAP**

**Scenario:**  
Using Citizen AI to ask questions related to government services, track public sentiment, and provide real-time assistance through an intelligent civic chatbot.

| **Steps** | **What does the user typically experience?** | **Interactions** | **Things (Digital touchpoints / physical objects)** | **Places** | **Goals & Motivations** | **Positive moments** | **Negative moments** | **Areas of opportunity** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Discover Citizen AI | Learns about Citizen AI through campaigns or social media | Browsing site, interacting with preview/demos | Project website, promotional post, banner links | Home/Public | Learn how to ask civic questions and get help easily | Curious about AI for civic issues | Unsure about credibility or privacy | Add demo videos, showcase use cases |
| Ask a question | Types a question about government schemes or services | Input into chatbox, press send | Streamlit chat UI, Flask backend | Home/Cyber Café | Get quick, accurate answers about policies or schemes | AI responds fast and relevantly | Response may miss specific context | Add location-aware or department-specific filters |
| View sentiment analysis | Submits feedback or reads trends about public issues | Clicks sentiment tab, sees charts | Streamlit dashboard | Office/Home | Understand how others feel about services | Visual graphs showing real trends | Feels the analysis is too generic | Add per-topic sentiment drill-down |
| Monitor dashboard | Government officer checks usage trends | Opens dashboard, checks metrics | Dashboard backend, JSON logs | Govt Office | See public opinion to improve services | Insightful data for policy change | Data delay or missing entries | Enable export, filters, and real-time streaming |
| Personalize AI behavior | Admin configures model (tone, verbosity) | Adjusts settings panel | Streamlit settings page | Admin Panel | Make responses formal/informal per use-case | Easy toggling of models | Unsure how changes affect output | Preview response type before applying |
| API fallback | IBM model fails, Groq takes over seamlessly | Auto-handled internally | Flask API, Groq API | Server | Keep system always responsive | Still receives an answer | Slower response time | Notify users about fallback status |
| Store feedback | Interaction stored automatically | No interaction — happens in background | JSON file system | Server | Help improve future service and analysis | Logged without interrupting flow | Not aware feedback is being saved | Add option to view/export user session data |
| Use on different devices | Citizen accesses app on desktop or mobile | Accesses via web browser | Responsive Streamlit UI | Anywhere | Flexibility in when/where to ask | Works across devices | Not fully mobile-optimized | Add mobile app version or PWA support |