Possible research categories with less important subjects marked blue

1. **Backend Implementation (Python):**
   * **Research Natural Language Processing (NLP):**
     + Study the fundamentals of NLP to understand how chatbots process and generate natural language text.
   * **Select an NLP Framework or Library:**
     + Research popular NLP frameworks and libraries in Python, such as NLTK, spaCy, or Hugging Face Transformers. Choose one that fits requirements.
   * **API Integration:**
     + Research API documentation and how to make API requests from Python.
   * **Database and Data Storage:**
     + Determine the need for a database to store chatbot responses, user interactions, or other data. Research databases like PostgreSQL, MySQL, or NoSQL options like MongoDB.
   * **User Authentication:**
     + Research methods for user authentication and authorization if chatbot requires it. Common approaches include OAuth2, API keys, or custom token-based systems.
   * **Backend Frameworks:**
     + Explore Python web frameworks such as Flask, Django, FastAPI, or Tornado for building backend server. Select the one that aligns with project's requirements.
   * **Development and Testing:**
     + Learn Python best practices for building robust and scalable web applications. Research how to create RESTful or GraphQL APIs for communication between the frontend and backend.
2. **Deployment on a Server:**
   * **Server Hosting Platforms:**
     + Research cloud hosting platforms like AWS, Google Cloud, Azure, or Heroku to deploy backend application. Consider factors like cost, scalability, and ease of use.
   * **Server Configuration:**
     + Learn about configuring and securing servers. Research server maintenance, including setting up firewalls, SSL certificates, and regular updates.
   * **Containerization:**
     + Explore containerization technologies such as Docker and container orchestration tools like Kubernetes if plan to use them to manage server.
   * **Deployment Automation:**
     + Research deployment automation tools like Ansible, Chef, or Docker Compose to streamline server setup and application deployment.
   * OR COULD USE HEROKU
3. **Frontend for the Main Website:**
   * **Web Development Basics:**
     + Familiarize with HTML, CSS, and JavaScript
   * **Web Frameworks and Libraries:**
     + Research frontend web frameworks and libraries like React, Vue.js, or Angular if plan to use a framework for building the user interface.
   * **Web Design Principles:**
     + Study web design principles, including responsive design, user experience (UX), and accessibility to create an engaging and user-friendly website.
   * **API Integration:**
     + Learn how to make API requests from the frontend using JavaScript. Explore methods like the Fetch API or third-party libraries like Axios.
   * **Deployment:**
     + Research how to deploy a static website or a frontend application to a web server. Explore CDNs (Content Delivery Networks) for faster content delivery.