**🧭 Overall Timeframe: 6–8 Weeks**

This assumes:

* You’re working part-time or as a small team
* You want a **working prototype** deployed to your website by the end

**📊 Phased Timeline and Task Breakdown**

**✅ PHASE 1: Planning & Requirement Analysis**

🕐 **Time**: 3–5 days  
📌 **Priority**: Critical  
🎯 **Goal**: Understand what to build and why

| **Task** | **Description** | **Output** |
| --- | --- | --- |
| Define user personas | Identify types: Donors, Volunteers, Refugees | User journey scenarios |
| Collect FAQs & queries | List real user questions | Base dataset for intents |
| Identify core intents | Categorize: donate, volunteer, contact, learn | Intent taxonomy |
| Choose backend & model | Pick: Flask vs FastAPI, spaCy + transformers | Stack decision |
| Write design doc intro | Purpose, scope, objectives, problem-solution | Start chatbot spec |

**✅ PHASE 2: NLP & Bot Architecture Design**

🕐 **Time**: 5–7 days  
📌 **Priority**: Very High  
🎯 **Goal**: Build the brain of the bot

| **Task** | **Description** | **Output** |
| --- | --- | --- |
| Define intents/entities | Annotate questions into categories | NLP schema |
| Write training data | Format examples in spaCy format | train.json or .spacy file |
| Build spaCy intent classifier | Use TextCategorizer pipeline | Trained NLP model |
| Design response handler | Rule-based + transformer fallback | Python class/function |
| Build conversation flow diagrams | Optional visual (XMind, Draw.io) | Flowchart PDF or image |

**✅ PHASE 3: Backend API + Bot Engine**

🕐 **Time**: 7–10 days  
📌 **Priority**: High  
🎯 **Goal**: Make it functional and testable locally

| **Task** | **Description** | **Output** |
| --- | --- | --- |
| Build Flask/FastAPI app | Basic REST API to send/receive messages | app.py or main.py |
| Integrate spaCy model | Parse message, detect intent | NLP response |
| Add Hugging Face transformer | Use distilgpt2 or similar | Generative reply |
| Add fallback logic | If spaCy fails, use GPT | Hybrid decision engine |
| Enable logging | Save inputs/outputs to JSON or DB | For analysis later |

**✅ PHASE 4: Frontend Integration**

🕐 **Time**: 5–7 days  
📌 **Priority**: Medium  
🎯 **Goal**: Make it visible on your website

| **Task** | **Description** | **Output** |
| --- | --- | --- |
| Build web chat widget | Use open source or JS snippet | HTML/JS frontend |
| Connect to API | JS sends message → Flask → response | Chat works live |
| Style for brand | Match Unity to Serve design | Custom CSS |
| Embed on site | Add widget or iframe to your HTML | Live chatbot! |

**✅ PHASE 5: Lead Capture + Multilingual (Optional Add-ons)**

🕐 **Time**: 7–10 days  
📌 **Priority**: Medium-to-Low (for v1)  
🎯 **Goal**: Expand capability & inclusiveness

| **Task** | **Description** | **Output** |
| --- | --- | --- |
| Capture user info | Store name, email with consent | Database or Google Sheets |
| Send auto email | Thank you or more info | SMTP integration |
| Add translation layer | Translate text (English ↔ Dari) | Google Translate API or bilingual model |
| Test with refugee users | Feedback loop | User insight report |

**✅ PHASE 6: Testing, Feedback, and Launch**

🕐 **Time**: 4–6 days  
📌 **Priority**: High  
🎯 **Goal**: Finalize and deploy

| **Task** | **Description** | **Output** |
| --- | --- | --- |
| Unit testing | Each NLP intent, response function | Test scripts |
| Functional testing | Complete chat flows | Bug fixes |
| Stakeholder feedback | Show to board or test users | Adjust wording/responses |
| Final deployment | Push to production server | Live site chatbot |
| Monitor metrics | Use basic analytics (e.g. Matomo) | Usage dashboard |

**📦 Optional: Weekly View Example**

| **Week** | **Goal** |
| --- | --- |
| **Week 1** | Requirements, FAQ analysis, design document |
| **Week 2** | NLP training (spaCy), chatbot architecture |
| **Week 3** | Backend API (Flask/FastAPI), intent-response engine |
| **Week 4** | Hugging Face integration, fallback responses |
| **Week 5** | Frontend web widget, embed on UnityToServe.org |
| **Week 6** | Feedback, testing, fixes, first launch |
| **Week 7–8** | Add-ons: email collection, multilingual replies, analytics |

**✅ Summary: Scientific Start-to-Finish Flow**

css

CopyEdit

[User Needs] → [Problem Framing] → [Intent Analysis] → [NLP Design] →

[Bot Engine Development] → [Website Integration] → [Feedback + Testing] → [Deployment + Metrics]