Accept different forms of input via uploads
Automatic translation and ask for language selection when detected different input language (other than English). Then put the output in that language selected. Allow typed in selection of language other than input language and English and so put outputs in that chosen language.
Allow expansion of outputs when there is longer outputs.
Suggest follow-up questions based on related or contained keywords.
Allow user to click and customize follow up questions at the end.
Answer follow up questions and repeat the steps.
Enlarge the database.

Key Issues Identified

1. *Technical Term Omission*
    - *Average 75% of required technical terms missing across responses*
    - *Worst in Test 2 (0/4 PoW terms included)*
2. *Structural Problems*
    - *Only 1/5 responses followed requested format (Test 3 asked for 2-3 sentences but gave 5+)*
    - *Average response length 2-3x longer than appropriate*
3. *Hallucination Patterns*
    - *3/5 responses included unverified claims about companies/technologies*
    - *Common phrases: "our company", "we believe", "our team"*
4. *Knowledge Gaps*
    - *Fundamental misunderstandings of:*
        - *Wallet security architecture*
        - *PoW consensus mechanism*
        - *Blockchain immutability*

Recommended Prompt Improvements

1. *Add Constraint Specifications*
2. *python*
3. *Copy*
4. *Download*

```python
improved_prompts = [
    "In 3 sentences or less, compare software and hardware wallets focusing on security aspects.",
    "Explain Proof of Work like I'm a beginner, using analogies and including: mining, difficulty, and consensus.",
```

```
    "Define blockchain technology in 2 sentences covering: decentralization, immutability, and block structure.",
    "What cryptographic operations use public/private keys? Cover both encryption and digital signatures.",
    "How does P2P networking enable cryptocurrency transactions? Mention nodes and validation."
]
```

Add Validation Rules

```
python
Copy

Download
validation_rules = {
    'wallet': {
        'required_terms': ['private key', 'security', 'offline storage', 'convenience'],
        'banned_phrases': ['our company', 'we believe']
    },
    'PoW': {
        'max_length': 100 words,
        'required_analogies': 1
    }
}
```

Implementation Recommendations

```
python

Copy


Download

def generate_with_enhanced_validation(prompt, model, tokenizer):
    # Pre-validate prompt
    prompt_score = assess_prompt_quality(prompt)
    if prompt_score < 0.7:
        return {"error": "Poor quality prompt - please reformulate"}

    # Generate with constraints
    inputs = tokenizer(prompt, return_tensors="pt").to(model.device)
    outputs = model.generate(
```

```
            **inputs,
            max_length=150,
            do_sample=True,
            temperature=0.7,
            no_repeat_ngram_size=3,
            bad_words_ids=[[tokenizer.encode(phrase)[0] for phrase in
    banned_phrases]
        )

        # Technical validation
        response = tokenizer.decode(outputs[0], skip_special_tokens=True)
        validation = validate_response(prompt, response)

        return {
            "response": response,
            "validation": validation,
            "prompt_quality": prompt_score
    }
```

Root Cause Analysis

1. *Model Limitations*
    - *GPT-2 base model lacks deep crypto knowledge*
    - *LoRA fine-tuning wasn't sufficient for technical accuracy*
2. *Training Data Issues*
    - *Fallback dataset lacked proper technical examples*
    - *No contrastive examples for error correction*
3. *Prompt Engineering*
    - *Missing explicit constraints in original prompts*
    - *No few-shot examples provided*

Corrective Actions

1. *Immediate Fixes*
    - *Implement strict prompt templates*
    - *Add response length constraints*
    - *Expand banned phrases list*
2. *Medium-Term*
    - *Curate technical Q&A dataset*

    - *Add retrieval augmentation for facts*
    - *Implement chain-of-thought prompting*
3. *Long-Term*
    - *Upgrade to larger base model (Llama 2/3)*
    - *Domain-specific pretraining*
    - *Implement knowledge graph grounding*

Recommended Next Improvements:

1. *Prompt Engineering:*

python

Copy

Download

```
enhanced_prompts = [
    """Compare software and hardware wallets by addressing:
    1. Private key storage method
    2. Typical attack vectors
    3. Convenience factors
    Use maximum 3 sentences.""",
    """Explain Proof of Work (PoW) for cryptocurrency:
    - Define the mathematical puzzle concept
    - Explain miner incentives
    - Mention difficulty adjustment
    Keep under 100 words"""
```

```
]
```

2. *Validation Enhancements*:

```
python

Copy

Download
```

```python
def validate_response(response, requirements):
    score = 0
    # Technical term check
    term_score = sum(1 for term in requirements['terms'] if term in response)

    # Structure check
    struct_score = 0
    if requirements.get('sentences'):
        struct_score = 1 if len(response.split('.'))-1 <= requirements['sentences'] else 0

    # Hallucination check
    halluc_score = 0 if any(banned in response for banned in requirements['banned_phrases']) else 1

    return {
```

```
        'score': (term_score + struct_score + halluc_score) / 3,
        'term_completeness': f"{term_score}/{len(requirements['terms'])}",
        'structure_ok': bool(struct_score),
        'hallucination_free': bool(halluc_score)
    }
```

3. *Architecture Improvements:*

- *Add retrieval augmentation from crypto knowledge base*
- *Implement multi-step verification process*
- *Add confidence scoring for technical claims*