```
import spacy

# Load the English language model for spaCy
nlp = spacy.load("en_core_web_sm")

# Define a function to preprocess text



def preprocess_text(text):
    # Process the text using the spaCy model
    doc = nlp(text)
    # Lemmatize the tokens, remove stop words, and filter out non-alphabetic tokens
    lemmatized_tokens = [token.lemma_.lower() for token in doc if not token.is_stop and token.is_alpha]
    # Return the list of lemmatized tokens
    return lemmatized_tokens

# Define a function to generate a shorter prompt
def generate_shorter_prompt(prompt):
    # Create a set to store important phrases
    important_phrases = set()
    # Process the input prompt using the spaCy model
    doc = nlp(prompt)
    # Extract important phrases using noun chunking and named entity recognition
    for chunk in doc.noun_chunks:
        important_phrases.add(chunk.text)
    for ent in doc.ents:
        important_phrases.add(ent.text)
    # Preprocess the important phrases using the preprocess_text function
    preprocessed_phrases = [' '.join(preprocess_text(phrase)) for phrase in important_phrases]
    # Remove duplicate phrases and combine the remaining phrases into a shorter prompt
    shorter_prompt = ' '.join(set(preprocessed_phrases))
    # Return the shorter prompt as a string
    return shorter_prompt


# Main function to test the generate_shorter_prompt function
if __name__ == "__main__":
    # Define an input prompt
    input_prompt = input("enter the promt:")
    # Generate a shorter prompt using the generate_shorter_prompt function
    shorter_prompt = generate_shorter_prompt(input_prompt)
    # Print the original prompt and the shorter prompt
    print("Original prompt:")
    print(input_prompt)
    print("\nShorter prompt:")
    print(shorter_prompt)
```