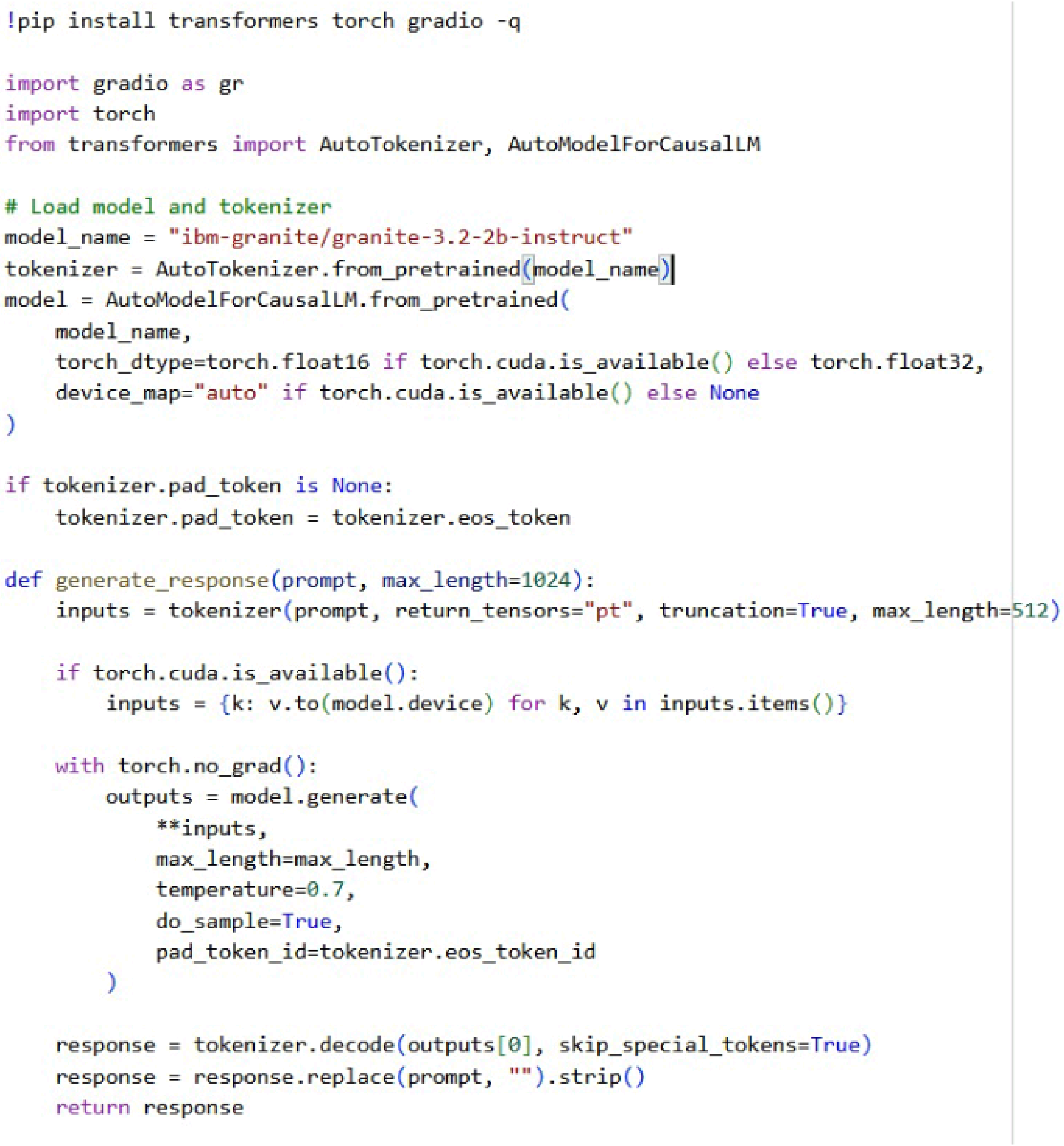
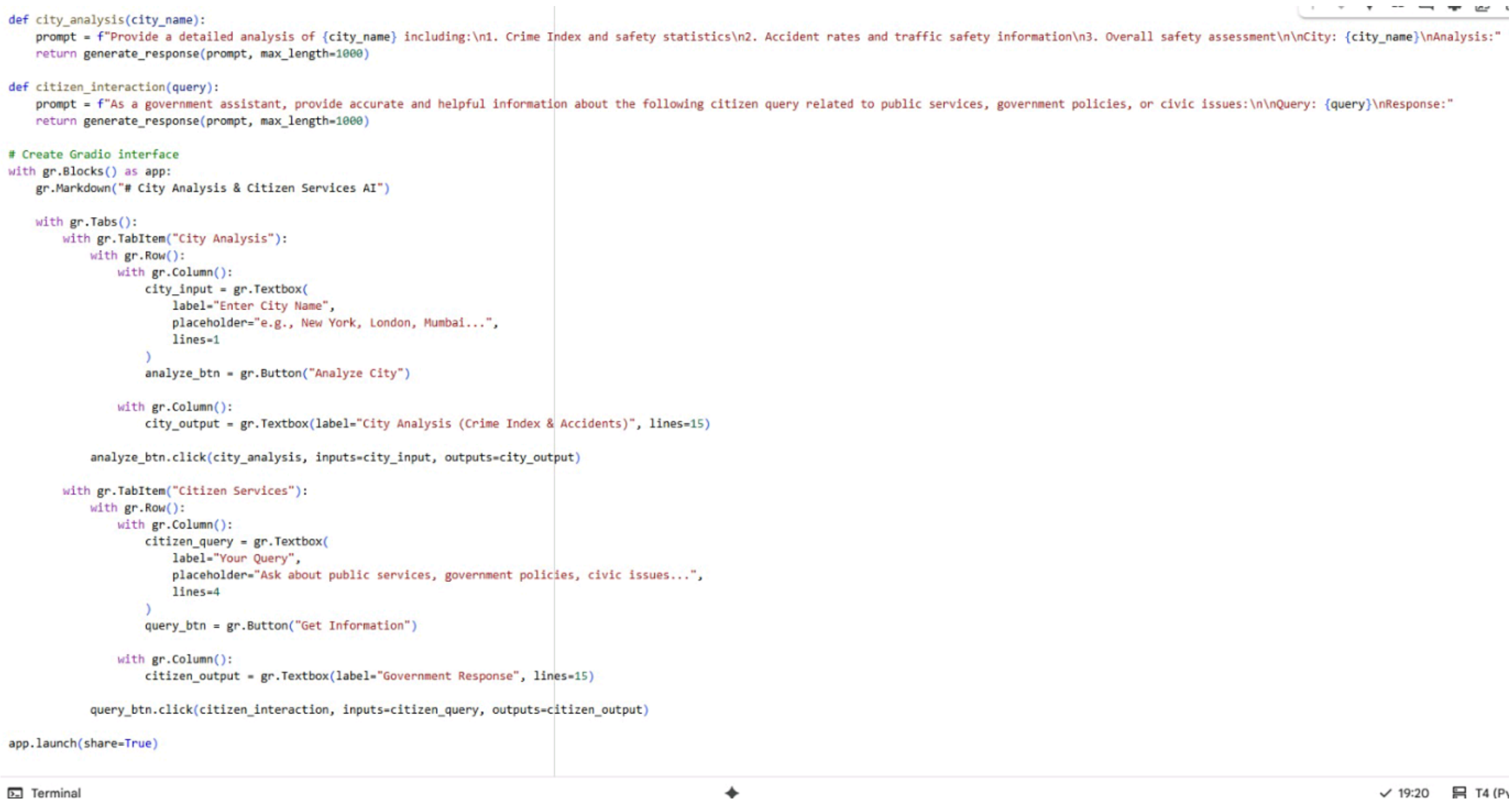
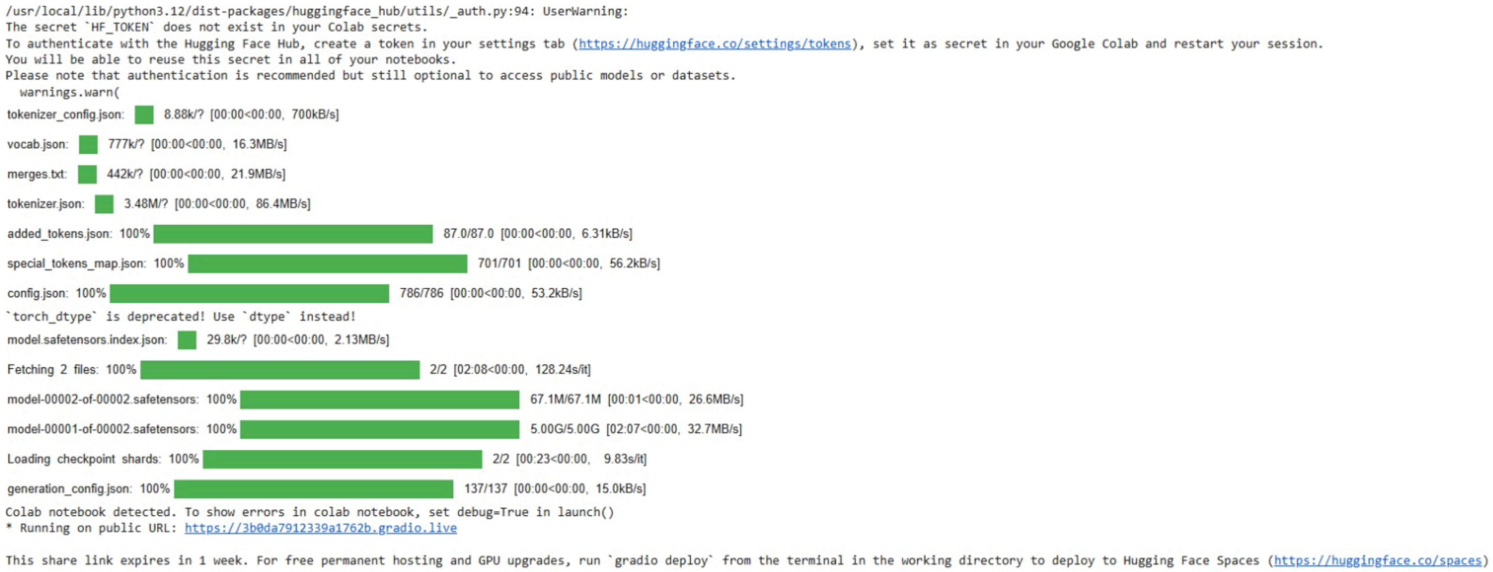


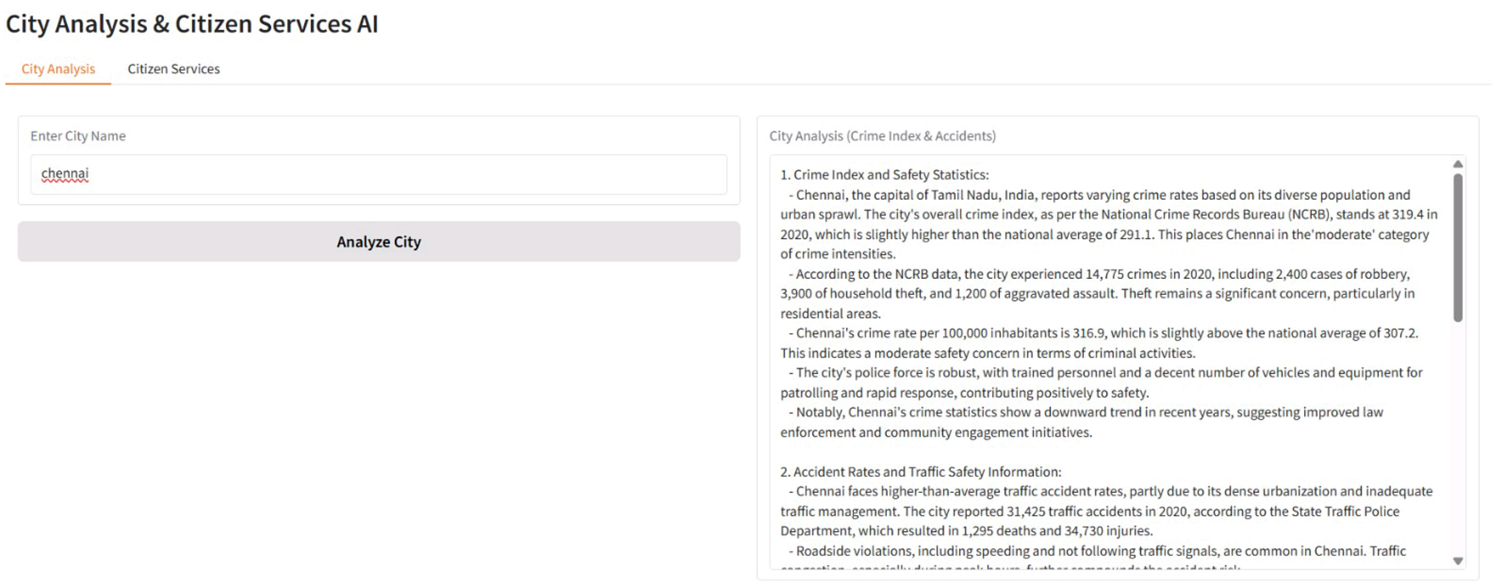
```
!pip install transformers torch gradio -q

import gradio as gr
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM

# Load model and tokenizer
model_name = "ibm-granite/granite-3.2-2b-instruct"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(
    model_name,
    torch_dtype=torch.float16 if torch.cuda.is_available() else torch.float32,
    device_map="auto" if torch.cuda.is_available() else None
)

if tokenizer.pad_token is None:
    tokenizer.pad_token = tokenizer.eos_token

def generate_response(prompt, max_length=1024):
    inputs = tokenizer(prompt, return_tensors="pt", truncation=True, max_length=512)

    if torch.cuda.is_available():
        inputs = {k: v.to(model.device) for k, v in inputs.items()}

    with torch.no_grad():
        outputs = model.generate(
            **inputs,
            max_length=max_length,
            temperature=0.7,
            do_sample=True,
            pad_token_id=tokenizer.eos_token_id
        )

    response = tokenizer.decode(outputs[0], skip_special_tokens=True)
    response = response.replace(prompt, "").strip()
    return response
```



```python
def city_analysis(city_name):
    prompt = f"Provide a detailed analysis of {city_name} including:\n1. Crime Index and safety statistics\n2. Accident rates and traffic safety information\n3. Overall safety assessment\n\nCity: {city_name}\nAnalysis:"
    return generate_response(prompt, max_length=1000)

def citizen_interaction(query):
    prompt = f"As a government assistant, provide accurate and helpful information about the following citizen query related to public services, government policies, or civic issues:\n\nQuery: {query}\nResponse:"
    return generate_response(prompt, max_length=1000)

# Create Gradio interface
with gr.Blocks() as app:
    gr.Markdown("# City Analysis & Citizen Services AI")

    with gr.Tabs():
        with gr.TabItem("City Analysis"):
            with gr.Row():
                with gr.Column():
                    city_input = gr.Textbox(
                        label="Enter City Name",
                        placeholder="e.g., New York, London, Mumbai...",
                        lines=1
                    )
                    analyze_btn = gr.Button("Analyze City")

                with gr.Column():
                    city_output = gr.Textbox(label="City Analysis (Crime Index & Accidents)", lines=15)

            analyze_btn.click(city_analysis, inputs=city_input, outputs=city_output)

        with gr.TabItem("Citizen Services"):
            with gr.Row():
                with gr.Column():
                    citizen_query = gr.Textbox(
                        label="Your Query",
                        placeholder="Ask about public services, government policies, civic issues...",
                        lines=4
                    )
                    query_btn = gr.Button("Get Information")

                with gr.Column():
                    citizen_output = gr.Textbox(label="Government Response", lines=15)

            query_btn.click(citizen_interaction, inputs=citizen_query, outputs=citizen_output)

app.launch(share=True)
```



```
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning: 
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
tokenizer_config.json:  8.88k/? [00:00<00:00, 700kB/s]
vocab.json:  777k/? [00:00<00:00, 16.3MB/s]
merges.txt:  442k/? [00:00<00:00, 21.9MB/s]
tokenizer.json:  3.48M/? [00:00<00:00, 86.4MB/s]
added_tokens.json: 100%  87.0/87.0 [00:00<00:00, 6.31kB/s]
special_tokens_map.json: 100%  701/701 [00:00<00:00, 56.2kB/s]
config.json: 100%  786/786 [00:00<00:00, 53.2kB/s]
`torch_dtype` is deprecated! Use `dtype` instead!
model.safetensors.index.json:  29.8k/? [00:00<00:00, 2.13MB/s]
Fetching 2 files: 100%  2/2 [02:08<00:00, 128.24s/it]
model-00002-of-00002.safetensors: 100%  67.1M/67.1M [00:01<00:00, 26.6MB/s]
model-00001-of-00002.safetensors: 100%  5.00G/5.00G [02:07<00:00, 32.7MB/s]
Loading checkpoint shards: 100%  2/2 [00:23<00:00,  9.83s/it]
generation_config.json: 100%  137/137 [00:00<00:00, 15.0kB/s]
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://3b0da7912339a1762b.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the working directory to deploy to Hugging Face Spaces (https://huggingface.co/spaces)
```

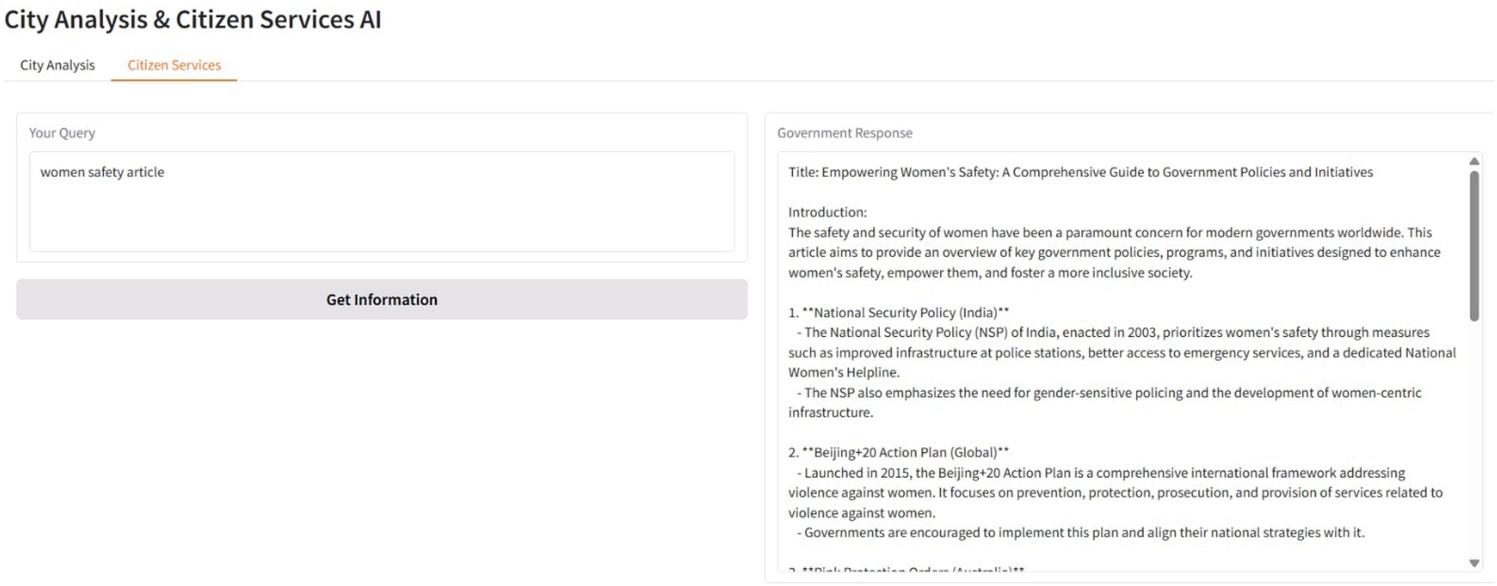


Enter City Name: chennai

City Analysis (Crime Index & Accidents)

1. Crime Index and Safety Statistics:
  - Chennai, the capital of Tamil Nadu, India, reports varying crime rates based on its diverse population and urban sprawl. The city's overall crime index, as per the National Crime Records Bureau (NCRB), stands at 319.4 in 2020, which is slightly higher than the national average of 291.1. This places Chennai in the'moderate' category of crime intensities.
  - According to the NCRB data, the city experienced 14,775 crimes in 2020, including 2,400 cases of robbery, 3,900 of household theft, and 1,200 of aggravated assault. Theft remains a significant concern, particularly in residential areas.
  - Chennai's crime rate per 100,000 inhabitants is 316.9, which is slightly above the national average of 307.2. This indicates a moderate safety concern in terms of criminal activities.
  - The city's police force is robust, with trained personnel and a decent number of vehicles and equipment for patrolling and rapid response, contributing positively to safety.
  - Notably, Chennai's crime statistics show a downward trend in recent years, suggesting improved law enforcement and community engagement initiatives.

2. Accident Rates and Traffic Safety Information:
  - Chennai faces higher-than-average traffic accident rates, partly due to its dense urbanization and inadequate traffic management. The city reported 31,425 traffic accidents in 2020, according to the State Traffic Police Department, which resulted in 1,295 deaths and 34,730 injuries.
  - Roadside violations, including speeding and not following traffic signals, are common in Chennai. Traffic congestion, especially during peak hours, further compounds the accident risk.



Your Query

women safety article

Government Response

Title: Empowering Women's Safety: A Comprehensive Guide to Government Policies and Initiatives

Introduction:
The safety and security of women have been a paramount concern for modern governments worldwide. This article aims to provide an overview of key government policies, programs, and initiatives designed to enhance women's safety, empower them, and foster a more inclusive society.

1. **National Security Policy (India)**
   - The National Security Policy (NSP) of India, enacted in 2003, prioritizes women's safety through measures such as improved infrastructure at police stations, better access to emergency services, and a dedicated National Women's Helpline.
   - The NSP also emphasizes the need for gender-sensitive policing and the development of women-centric infrastructure.

2. **Beijing+20 Action Plan (Global)**
   - Launched in 2015, the Beijing+20 Action Plan is a comprehensive international framework addressing violence against women. It focuses on prevention, protection, prosecution, and provision of services related to violence against women.
   - Governments are encouraged to implement this plan and align their national strategies with it.

3. **Pink Protection Orders (Australia)**