Interactive Storytelling Bot

Our team

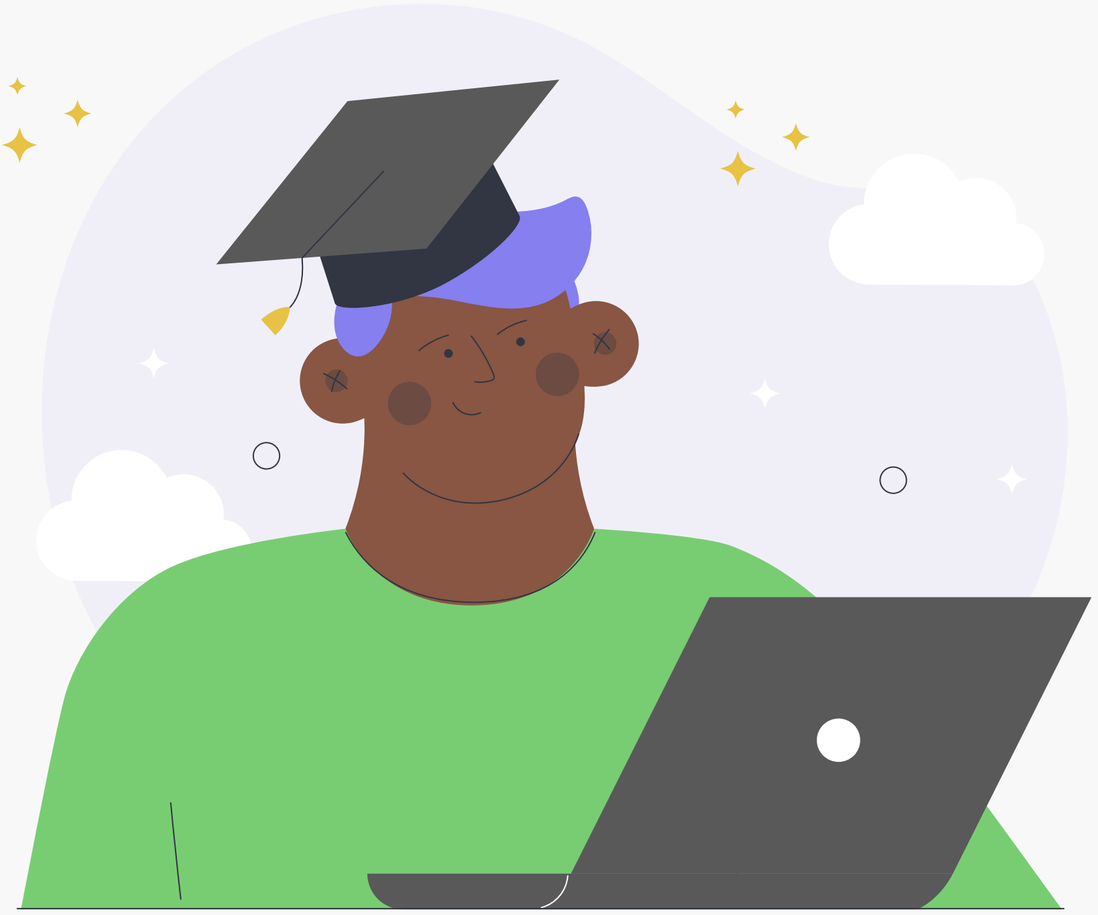
Omnia Mohamed

Habiba Khafage

Norhan Hossam

Hanan Elhosary

Table of contents



01

Introduction

Introduction

Problem Statement:

Many learners struggle with creative expression and story development. Current educational tools often lack interactive methods to encourage narrative thinking.

Motivation:

We chose this problem because storytelling can improve language skills, spark creativity, and make learning more engaging. Our goal was to create a tool that combines interactivity with AI to generate customized stories.

Introduction

Target Users:

- Students across various educational levels
- Teachers seeking interactive teaching aids
- Individuals interested in storytelling and writing practice

Expected Impact:

The solution encourages user participation in creative writing, improves engagement in educational settings, and enhances the storytelling experience using AI-generated content.

02

System Features (Demo)

System Features

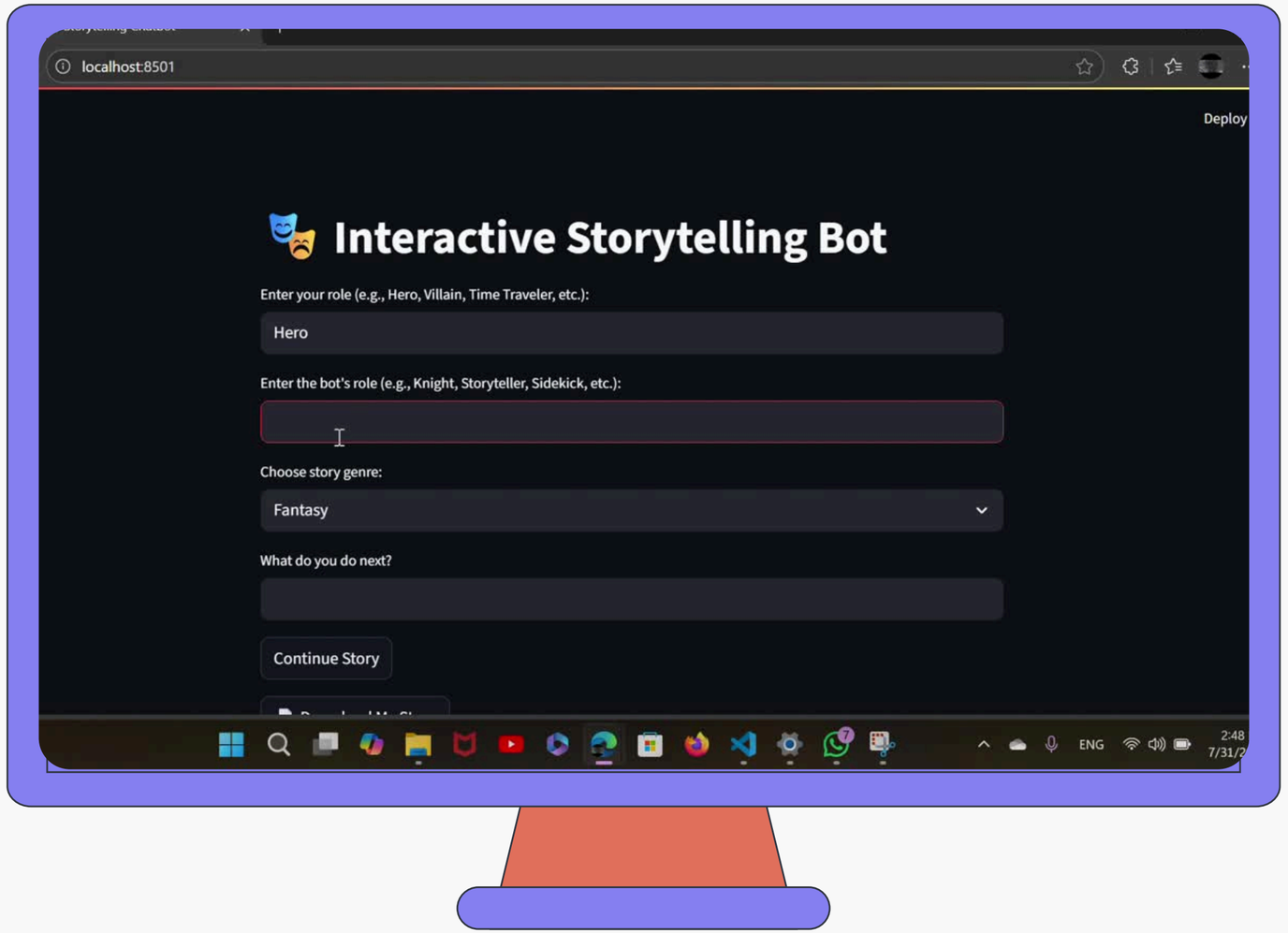
Core Features:

- Role and genre selection to personalize story context
- Real-time story continuation through AI
- Story progression based on user inputs
- Inventory tracking (e.g., "You found a magic key.") for enhanced interactivity
- Downloadable story export as PDF

System Features

Demo Options:

- Live walkthrough of app interface using Streamlit
- Screenshots highlighting:
- Role/genre input interface
- Story generation and interaction
- Inventory display
- PDF download button



System Features

Code Snippet (from app.py):

```
role = st.selectbox("Choose your role:", ["Hero", "Villain", "Detective"])
genre = st.selectbox("Choose story genre:", ["Fantasy", "Sci-Fi", "Mystery"])
...
res = requests.post("http://localhost:8000/story", json={
    "role": role,
    "genre": genre,
    "history": st.session_state.chat_history,
    "user_input": user_input,
    "inventory": st.session_state.inventory
})
```

03

Technical Architecture

Technical Architecture

Architecture Diagram

Frontend

Streamlit app for user interaction

Backend API

FastAPI server handling story generation

LLM Provider

Together API (e.g., Mixtral or LLaMA models)

Logic Layer

Handles story flow, inventory detection, PDF generation

Technical Architecture

Data Flow:

01
User selects role and genre

02
Input sent to FastAPI backend

03
Backend calls LLM API to generate story

04
Inventory items extracted and updated

05
Output displayed in frontend + option to export to PDF

04

Challenges Faced

Challenges Faced

1.Latency in API Calls:

- The story generation depends on external API (Together AI), which can occasionally be slow.
- Solution: Limited max tokens and added loading indicators on frontend.

2.Inventory Extraction Logic

- Parsing user story text to extract inventory items was non-trivial.
- Solution: Used regular expressions to match action phrases like “You found...” or “You picked up...”

Challenges Faced

3. Deployment and CORS Issues:

- Connecting the Streamlit frontend to FastAPI backend locally required enabling CORS.
- Solution: Added middleware to FastAPI for universal access.

Challenges Faced

Code Snippet (from story_logic.py):

```
pattern = r"(You (find|picked up|take|grab).*?[\.\!])"
matches = re.findall(pattern, text, flags=re.IGNORECASE)
```

05 Future Work

Future Work

Cloud Deployment:

Host the system on cloud (e.g., Azure or AWS) to make it publicly accessible.

Enhanced Interaction:

Add branching choices (multiple-choice actions), character avatars, and voice narration.

User Authentication:

Allow users to log in, save stories, and track progress over time.

Enhanced Interaction:

Support story generation in Arabic and other languages using multilingual LLMs.

05

Conclusion

Conclusion

What We Accomplished:

- Developed an end-to-end interactive storytelling platform
- Integrated frontend (Streamlit) with backend (FastAPI + LLM)
- Implemented logic for story progression, inventory tracking, and PDF export

What We Learned:

- Neptune is very far How to structure AI-based user interaction systems
- Effective use of APIs and prompt engineering for creative generation
- Practical deployment of ML models within interactive UIs

Real-World Usefulness:

This project demonstrates the potential of AI in education and entertainment by merging language models with storytelling to enhance user creativity.

Thanks