# Integration of Artificial Intelligence in Education and Software Development

Luna Schätzle
Florian Prandstetter

HTL Anichstraße, Department of Business Informatics
Thesis Supervisor:
Mag. Dr. Dipl. -Ing. Albert Greinöcker
MMag.a Eva-Maria Egger, MA

Diploma Thesis Defense – April 2025

# Introduction

- **Presenter:** Luna Schätzle – Project Lead (AI evaluation, backend & website)
- **Objective:** Open-source AI platform for education
- **Focus:** Evaluate various AI models for multiple use cases
- **Platform:** Enable students to access and experiment with AI
- **Motivation:** Overcome high resource requirements of current Open Source AI models

# Project Team and Management

- **Team Members:** Luna Schätzle, Florian Prandstetter
- **Project Coordination:** Regular meetings, discussions, and planning sessions
- **Tools Employed:**
  - GitHub for version control and collaborative coding
  - Discord for communication and coordination
  - Google Sheets for time tracking
  - LaTeX for comprehensive documentation

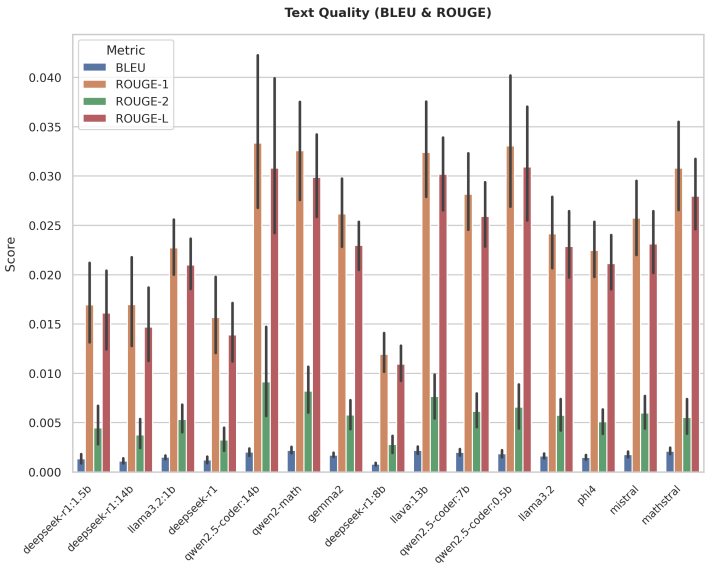
# Theoretical Background

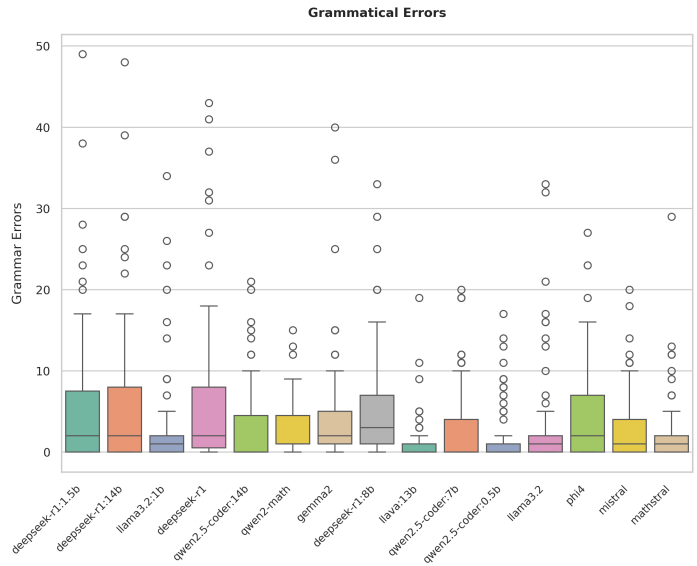
- **LLMs Integration:** Evaluation and incorporation of various Large Language Models.
- **Interfaces:** API connections, local models (e.g., Ollama), and OpenAI API.
- **Evaluation:** Systematic testing of open source models

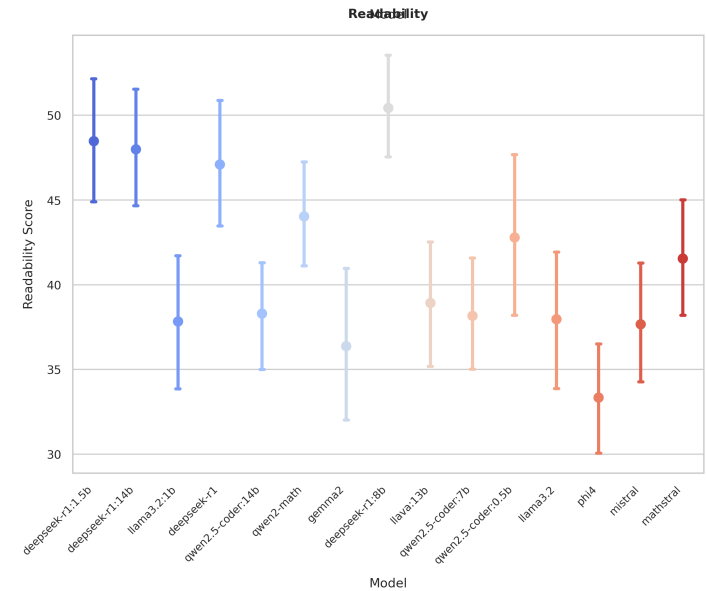
# Testing and Evaluation

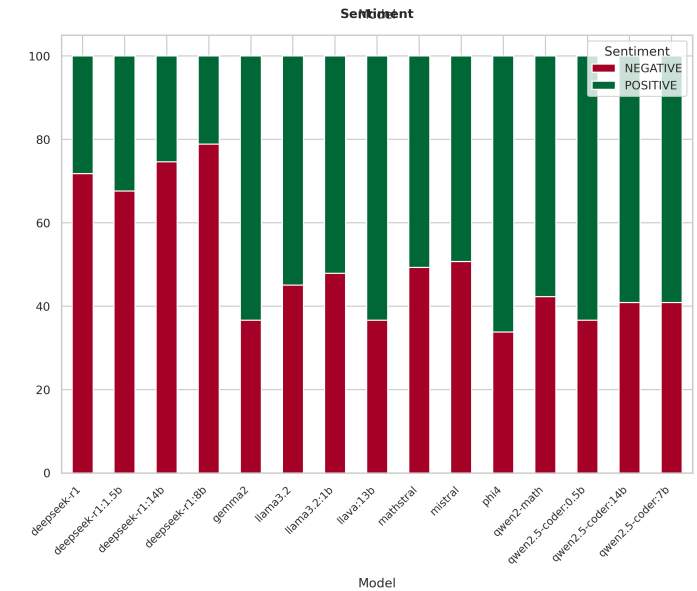
- Evaluation of models: Llama3.2, Deepseek-r1, gemma2, qwen, ...
- Testing methods: Different prompts and tasks where asked the models (automated via Python script)
- Evaluation criteria:
  - response time
  - accuracy
  - resource usage
  - BLEU score
  - readability
  - Textquality

# Evaluation Results: Quantitative metrics




# Evaluation Results: Qualitative metrics

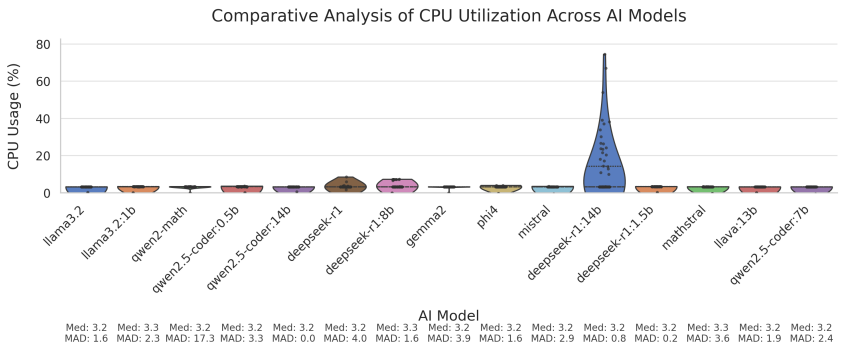



Figure: CPU Usage Comparison

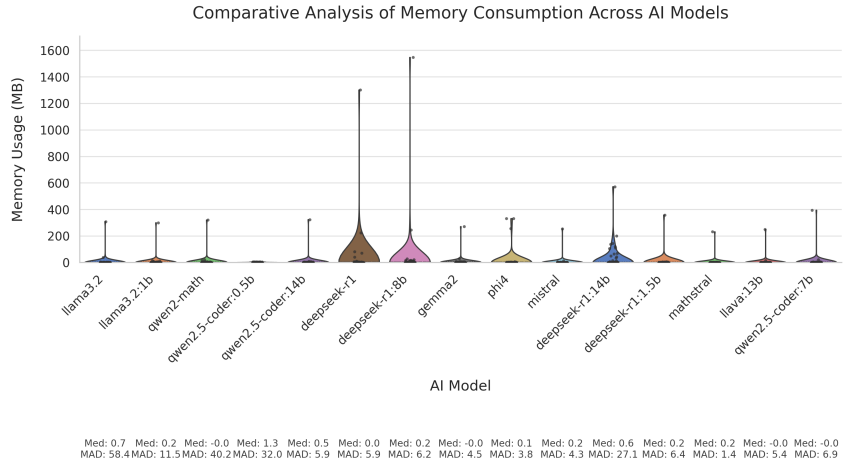


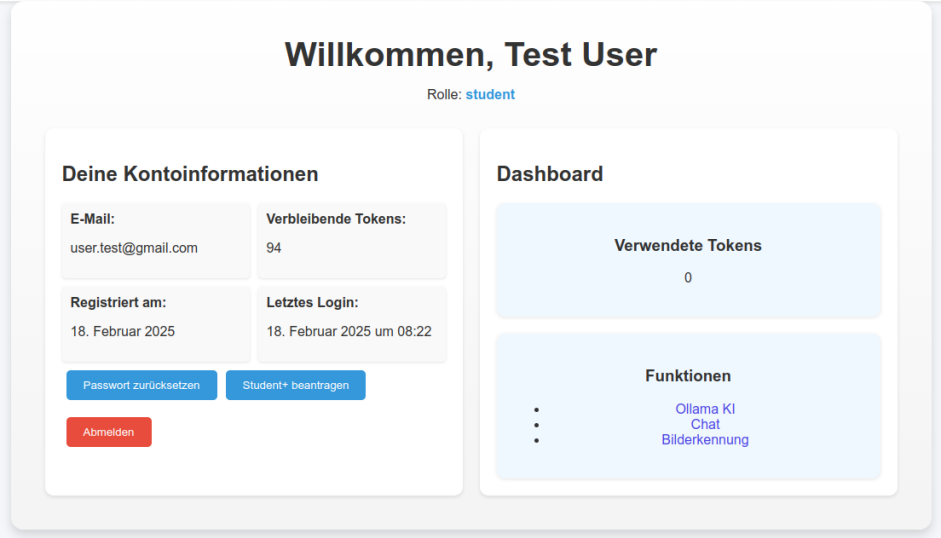

Figure: Memory Usage Comparison

# Website Platform

- Developed to make AI accessible to students.
- Built with:
  - Vue.js (Frontend)
  - Flask (Backend API)
  - Firebase (User data & authentication)
- Purpose: Central interface for interacting with various AI tools.

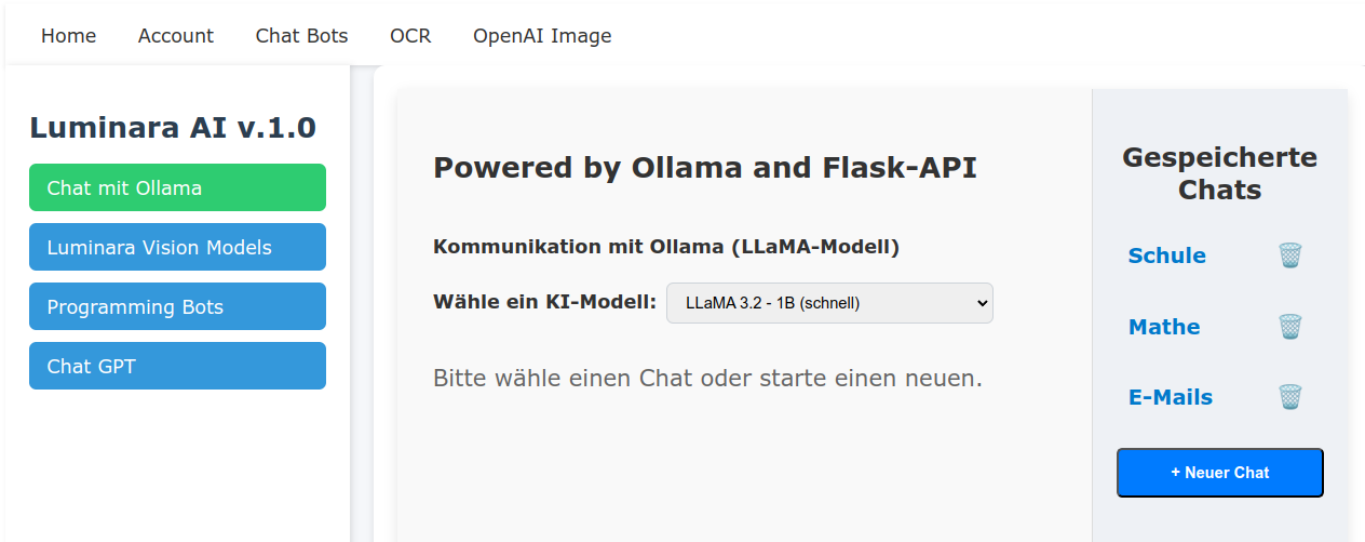
# User System

- Registration and secure login
- Profile management
- Firebase-based authentication




# Chatbot Interface

- Multiple AI models available via tabs:
  - ChatGPT (OpenAI API)
  - Local models (e.g., Ollama)
  - Programming Assistant
- Vision models: LLaVA, LLaMA 3.2 Vision

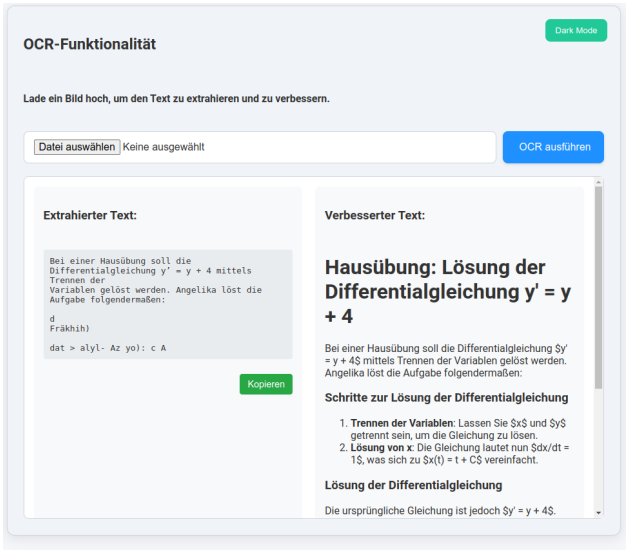



# Image Generation

- Generate images from text prompts
- Uses DALL·E (OpenAI) via backend API
- Fully integrated into frontend

# OCR and Image Recognition

- OCR with Tesseract
- Post-processing using LLaMA 3.2 Vision
- Recognize and explain content in uploaded images




# AI in Economics and Ethics

**Applications:**

- Customer service & support
- Supply chain management
- Predictive analytics
- Data analysis
- Process automation

# AI in Economics and Ethics

**Ethical & Social Concerns:**

- Bias in training data
- Transparency & accountability
- Privacy and data protection
- Impact on employment

# AI in Economics and Ethics

**Regulatory Challenges:**

- Inconsistent global regulations
- EU AI Act considerations [EUR-Lex: 2024/1689]
- Data security standards (e.g., GDPR [EUR-Lex: 2016/679])

# Open Source Overview

- **Definition:** Collaborative, transparent development with public source code.
- **Advantages:** Cost efficiency, flexibility, improved security through peer review, high compatibility.
- **Economic Impact:**
  - Drives innovation & cross-industry collaboration
  - Empowers startups and lowers entry barriers

# Challenges and Revenue Models

- **Challenges:** Fragmentation, limited support, licensing complexities, security risks.
- **Revenue Models:** Open core, managed services, support contracts, donations, dual licensing.
- **Our Approach:** Utilize open source tools (e.g., Python, Flask, Vue.js) under GNU GPL-3.0 for transparency and collaboration.

# Conclusion

- Summary of achievements
- Insights gained during the development
- Future potential of the system
- Final thoughts and acknowledgments

# Server

- **Server Hardware:**
  - CPU: Intel Core i5.8600k
  - GPU: NVIDA GeForce RTX 2060
  - RAM: 16GB DDR4
  - Motherboard: H370 Chipset
  - Power Supply: 500W BeQuiet
  - Storage: 512GB NVMe SSDd
- **Used Operating System:** The Server is running with the Ubuntu Server Operating System. The Operating System has been chosen due to the good cuda support.

# Server

- **Networking:**
  - Axios: Used for server requests
  - Tailscale: VPN tunnel used for secure remote access
- **Backup and Recovery:** Regular system backups have been made to avoid data loss.

# Flask Service

- Flask as a Web Framework
- Architecture and Service Structure
- Restful Endpoints and Functionalities
- Deployment with Docker

# Visual Studio Code Extension

- VS Code API / Typescript
- Server Request
- Integrated Chatbot
- Status Bar Item

# Operating System Market Share

- **Competitors:** Android, Microsoft Windows, Apple and Linux hold most of the market.
- Bild
- **For Servers:** When looking at Server Operating Systems specifically The main Competitors are Red Hat and Microsoft.
- Bild