innohassle.ru

от рус. «без хлопот» или «инно-суета»

search

find everything you need in Innopolis

problem?

problem.

searching for information in Innopolis takes effort 😮‍💨

problem

searching for information in Innopolis takes effort 😮 💨

following information sources may be considered:
- University and City documents
- Moodle, Eduwiki, Campus life, Outlook
- Dozens of Telegram channels

problem

searching for information in Innopolis takes effort 😮‍💨

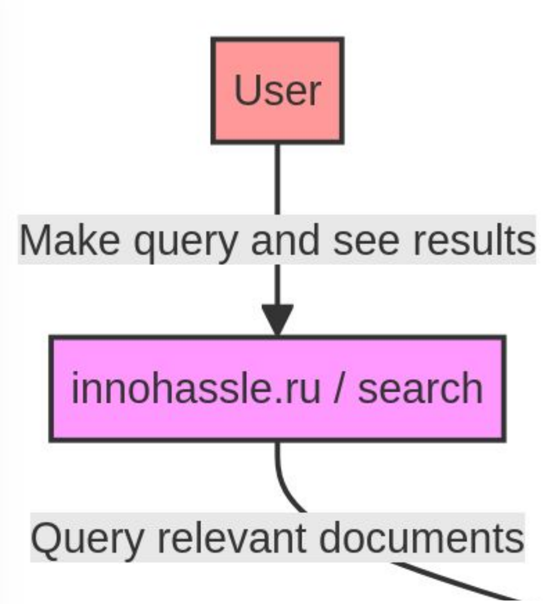
following information sources may be considered:
- University and City documents
- Moodle, Eduwiki, Campus life, Outlook
- Dozens of Telegram channels

for now we will focus on Moodle

solution / architecture

our solution consists of four parts:

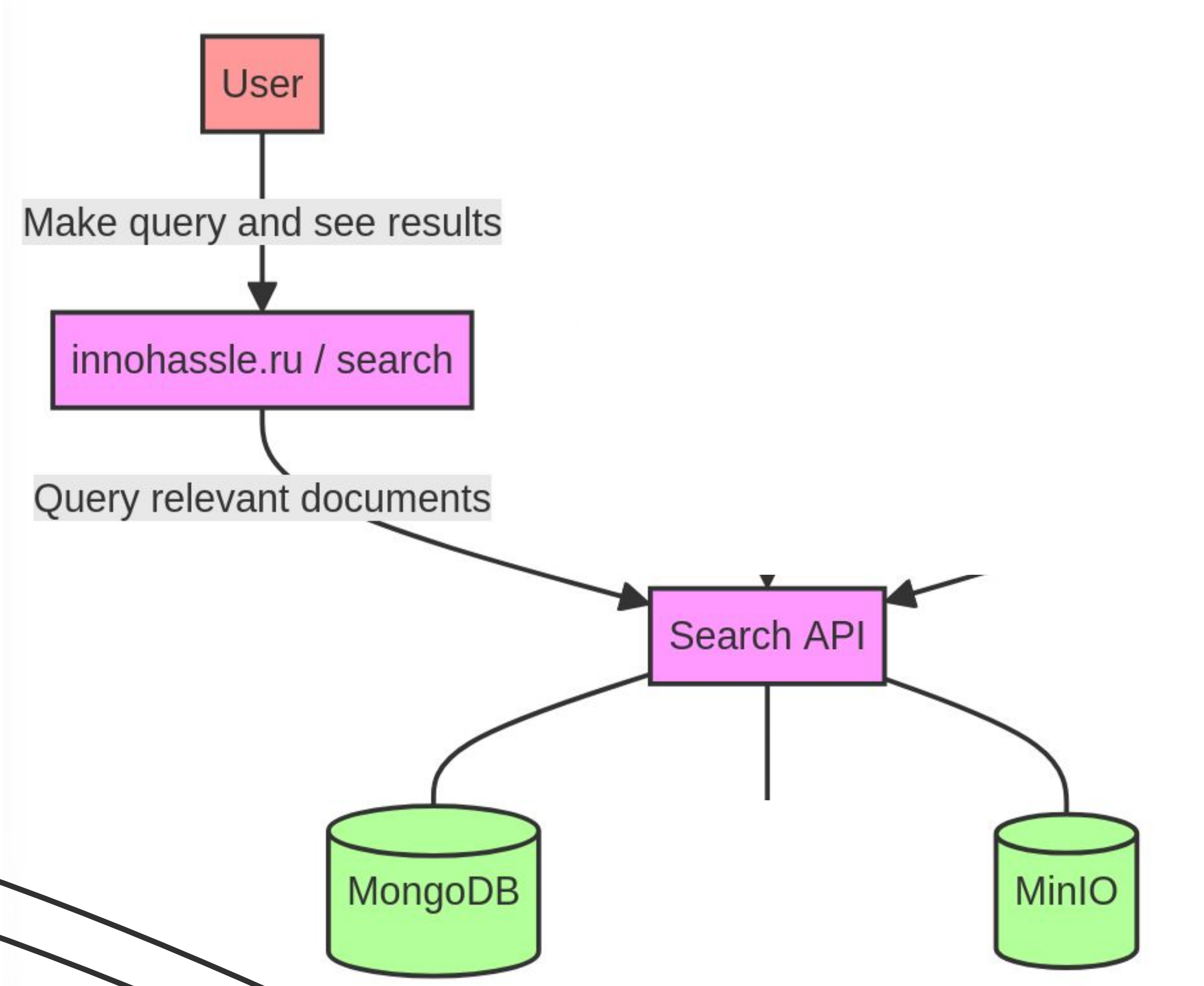
- Website



solution / architecture

our solution consists of four parts:

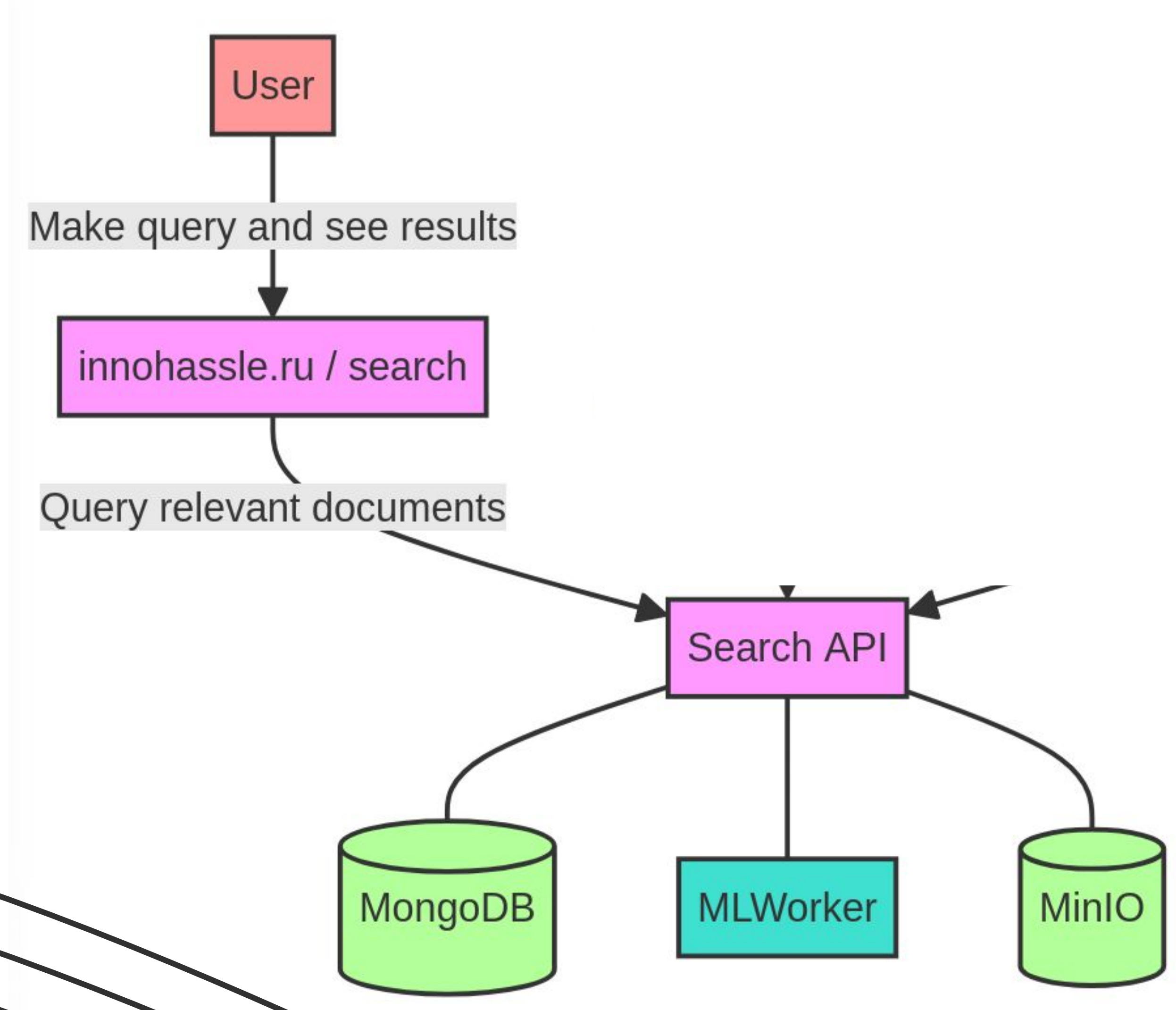
- Website
- Search API



solution / architecture

our solution consists of four parts:

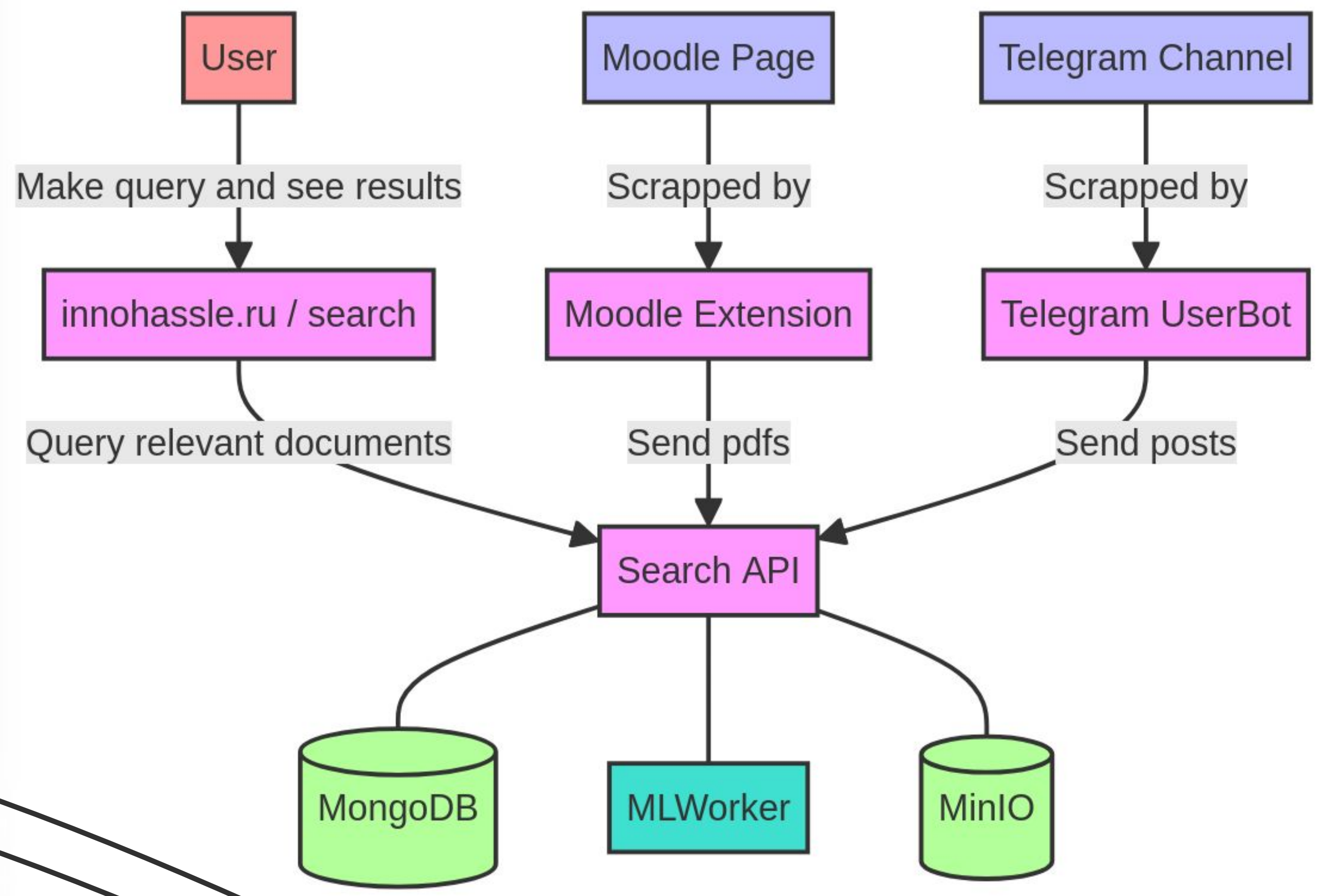
- Website
- Search API
- ML worker



solution / architecture

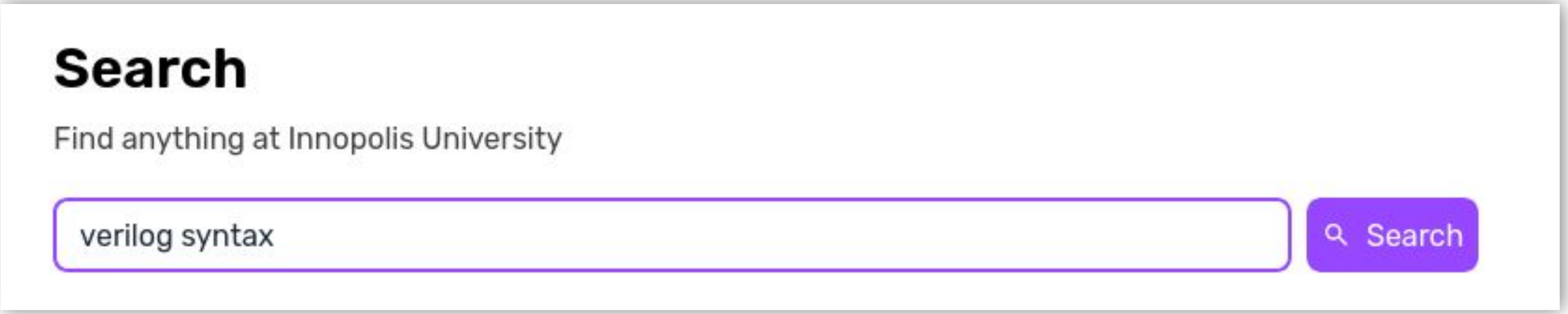
our solution consists of four parts:

- Website
- Search API
- ML worker
- Scrappers



solution / website

search bar



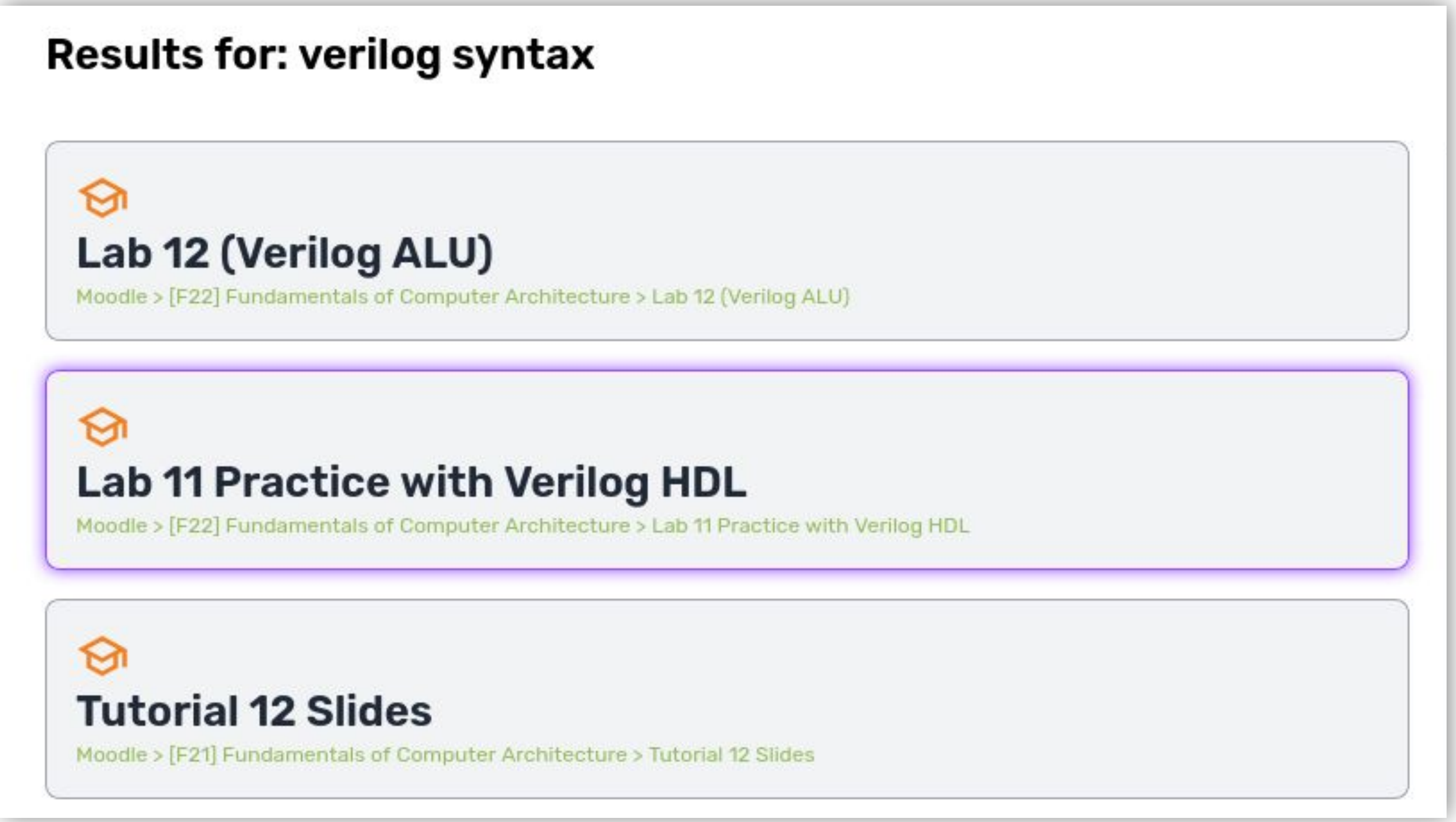
Search

Find anything at Innopolis University

verilog syntax

solution / website

result cards with specified source



Results for: verilog syntax

Lab 12 (Verilog ALU)

Moodle > [F22] Fundamentals of Computer Architecture > Lab 12 (Verilog ALU)

Lab 11 Practice with Verilog HDL

Moodle > [F22] Fundamentals of Computer Architecture > Lab 11 Practice with Verilog HDL

Tutorial 12 Slides

Moodle > [F21] Fundamentals of Computer Architecture > Tutorial 12 Slides

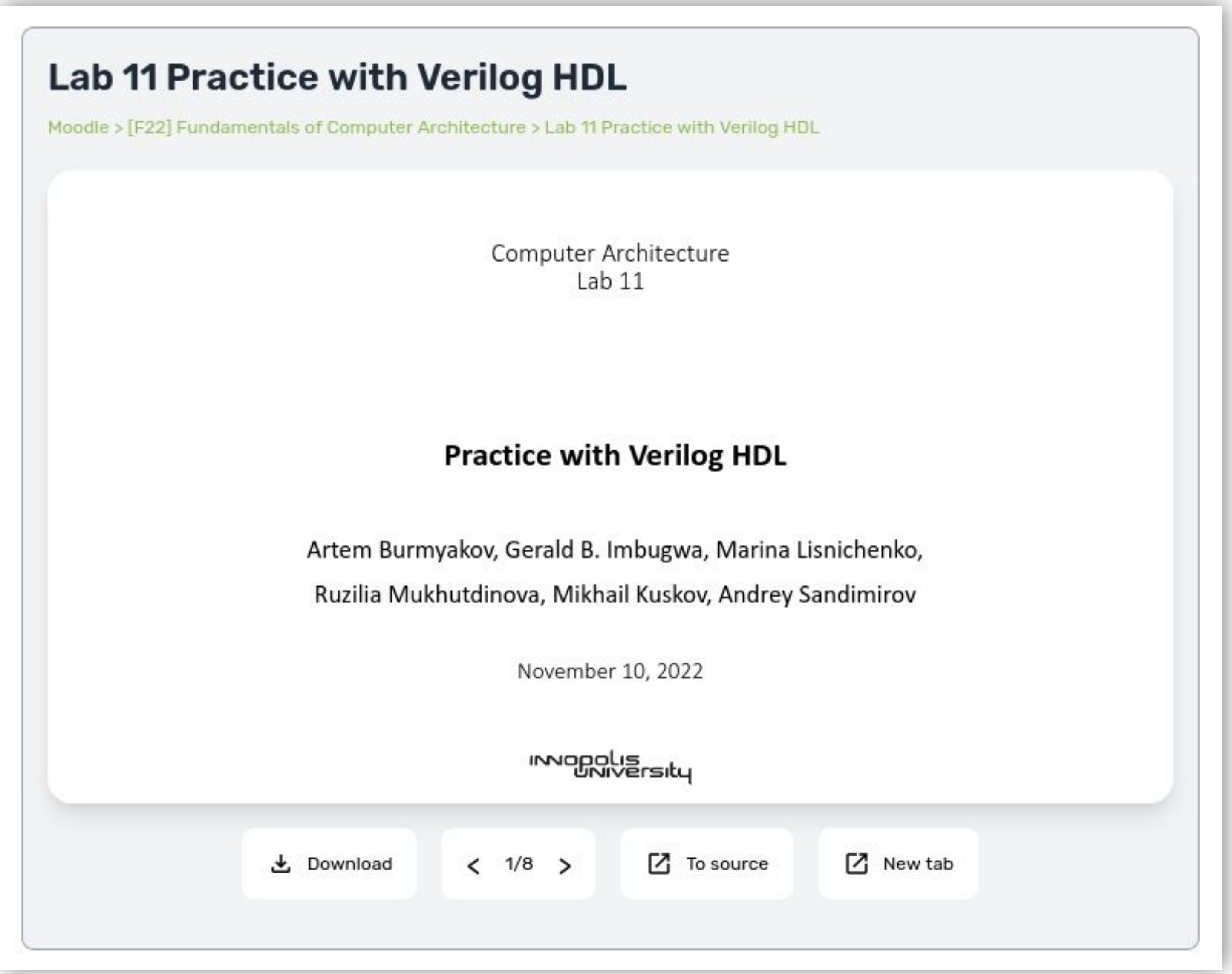
solution / website

preview of PDF documents

preview of Telegram posts

link to download the file

link to original source



Lab 11 Practice with Verilog HDL

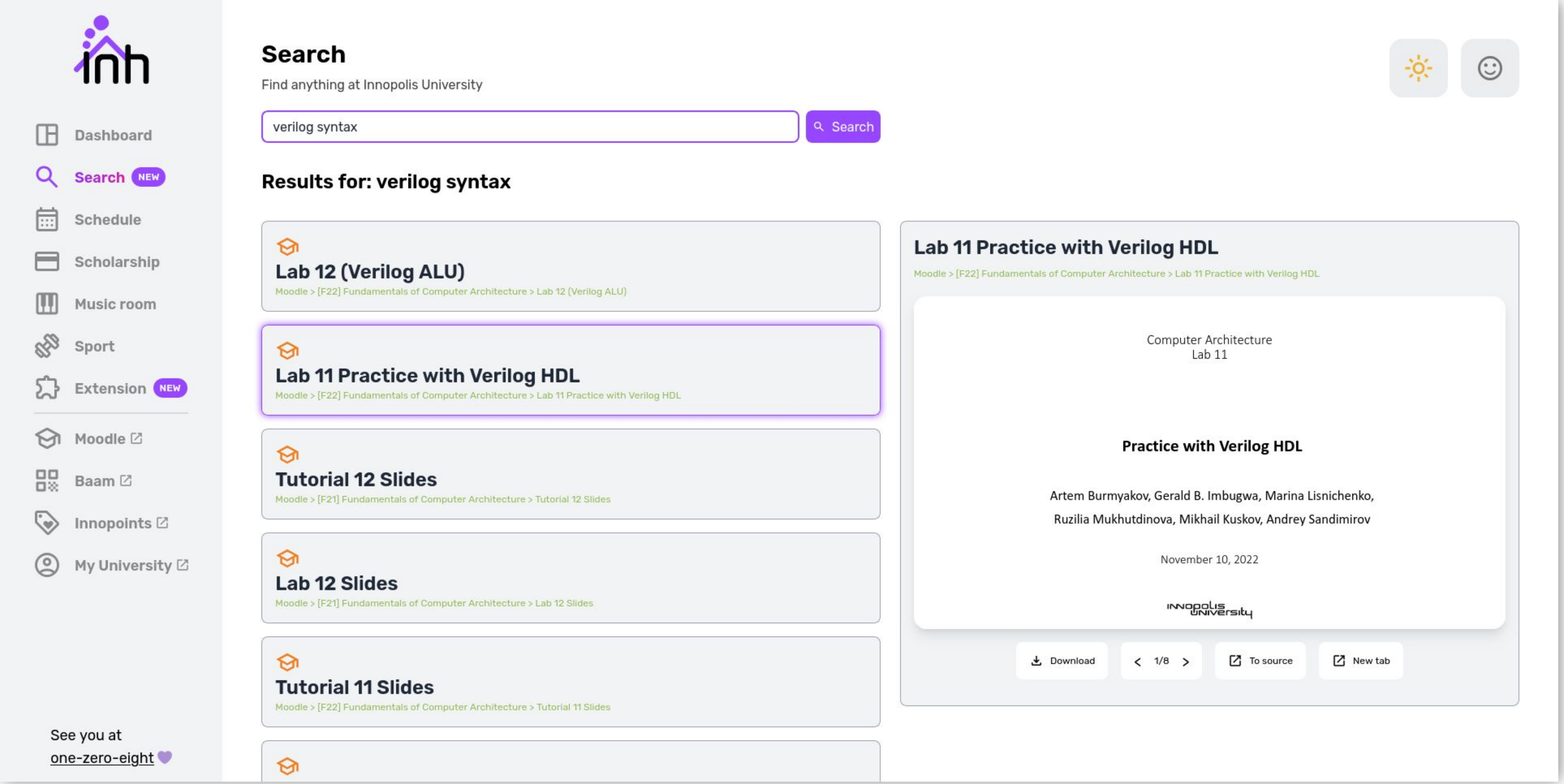
Moodle > [F22] Fundamentals of Computer Architecture > Lab 11 Practice with Verilog HDL

Computer Architecture
Lab 11

Practice with Verilog HDL

Artem Burmyakov, Gerald B. Imbugwa, Marina Lisnichenko,
Ruzilia Mukhutdinova, Mikhail Kuskov, Andrey Sandimirov

November 10, 2022



See you at one-zero-eight

Search

Find anything at Innopolis University

verilog syntax

Results for: verilog syntax

Lab 12 (Verilog ALU)
Moodle > [F22] Fundamentals of Computer Architecture > Lab 12 (Verilog ALU)

Lab 11 Practice with Verilog HDL
Moodle > [F22] Fundamentals of Computer Architecture > Lab 11 Practice with Verilog HDL

Tutorial 12 Slides
Moodle > [F21] Fundamentals of Computer Architecture > Tutorial 12 Slides

Lab 12 Slides
Moodle > [F21] Fundamentals of Computer Architecture > Lab 12 Slides

Tutorial 11 Slides
Moodle > [F22] Fundamentals of Computer Architecture > Tutorial 11 Slides

Lab 11 Practice with Verilog HDL
Moodle > [F22] Fundamentals of Computer Architecture > Lab 11 Practice with Verilog HDL

Computer Architecture
Lab 11

Practice with Verilog HDL

Artem Burmyakov, Gerald B. Imbugwa, Marina Lisnichenko,
Ruzilia Mukhutdinova, Mikhail Kuskov, Andrey Sandimirov

November 10, 2022

Innopolis University

website / stack

this technologies encourage reusability and better organization of code

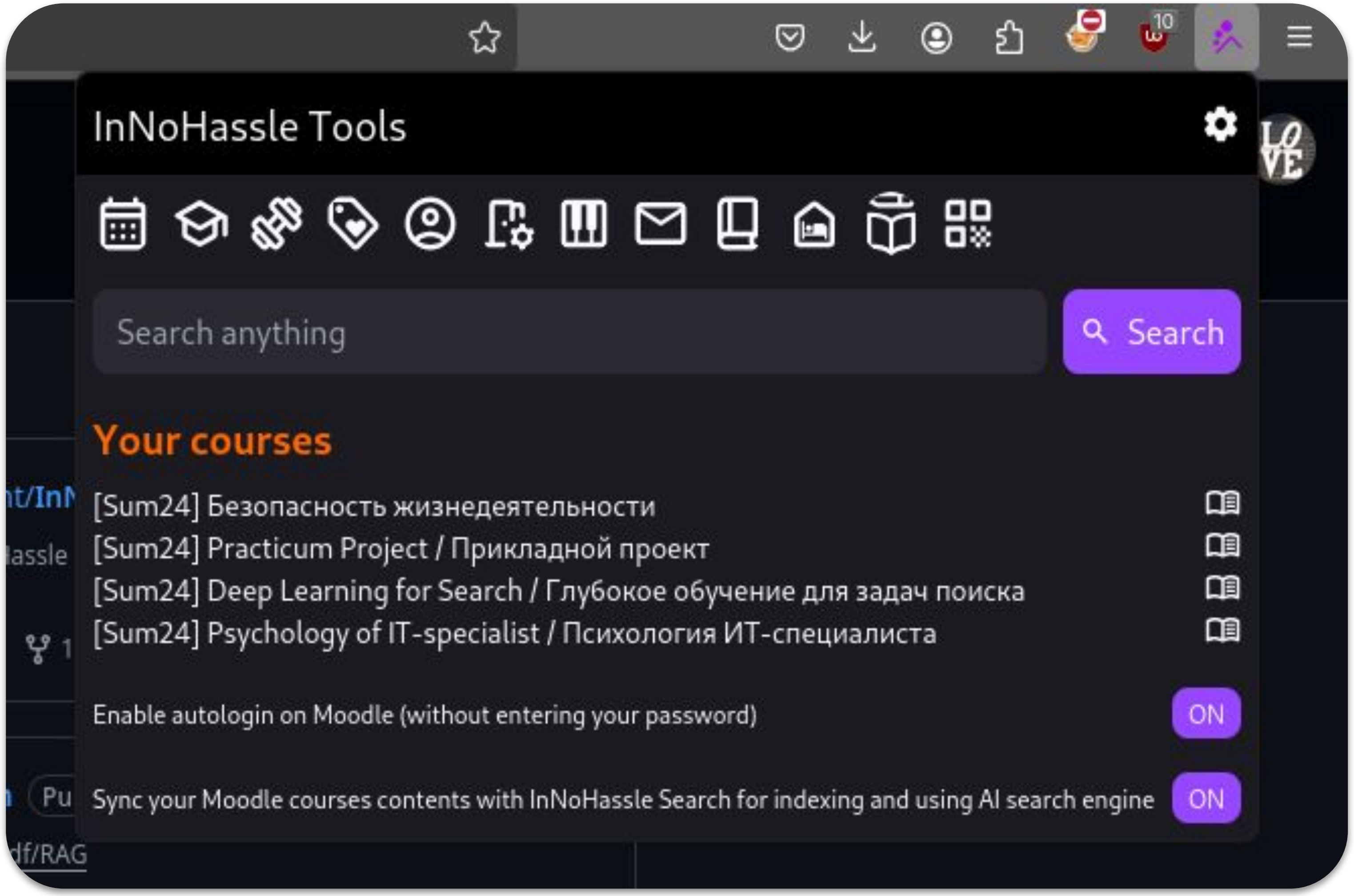
- Node.js
- TypeScript
- React & Next.js
- Tailwind CSS, Iconify – Styling
- Axios, React Query – Data fetching

solution / extension

Browser Extension:

- extend moodle session
- quick links to resources
- fast access to study materials
- integrated search engine

- optionally send files to index



Your courses

[Sum24] Безопасность жизнедеятельности

[Sum24] Practicum Project / Прикладной проект

[Sum24] Deep Learning for Search / Глубокое обучение для задач поиска

[Sum24] Psychology of IT-specialist / Психология ИТ-специалиста

Enable autologin on Moodle (without entering your password) ON

Sync your Moodle courses contents with InNoHassle Search for indexing and using AI search engine ON

solution / backend

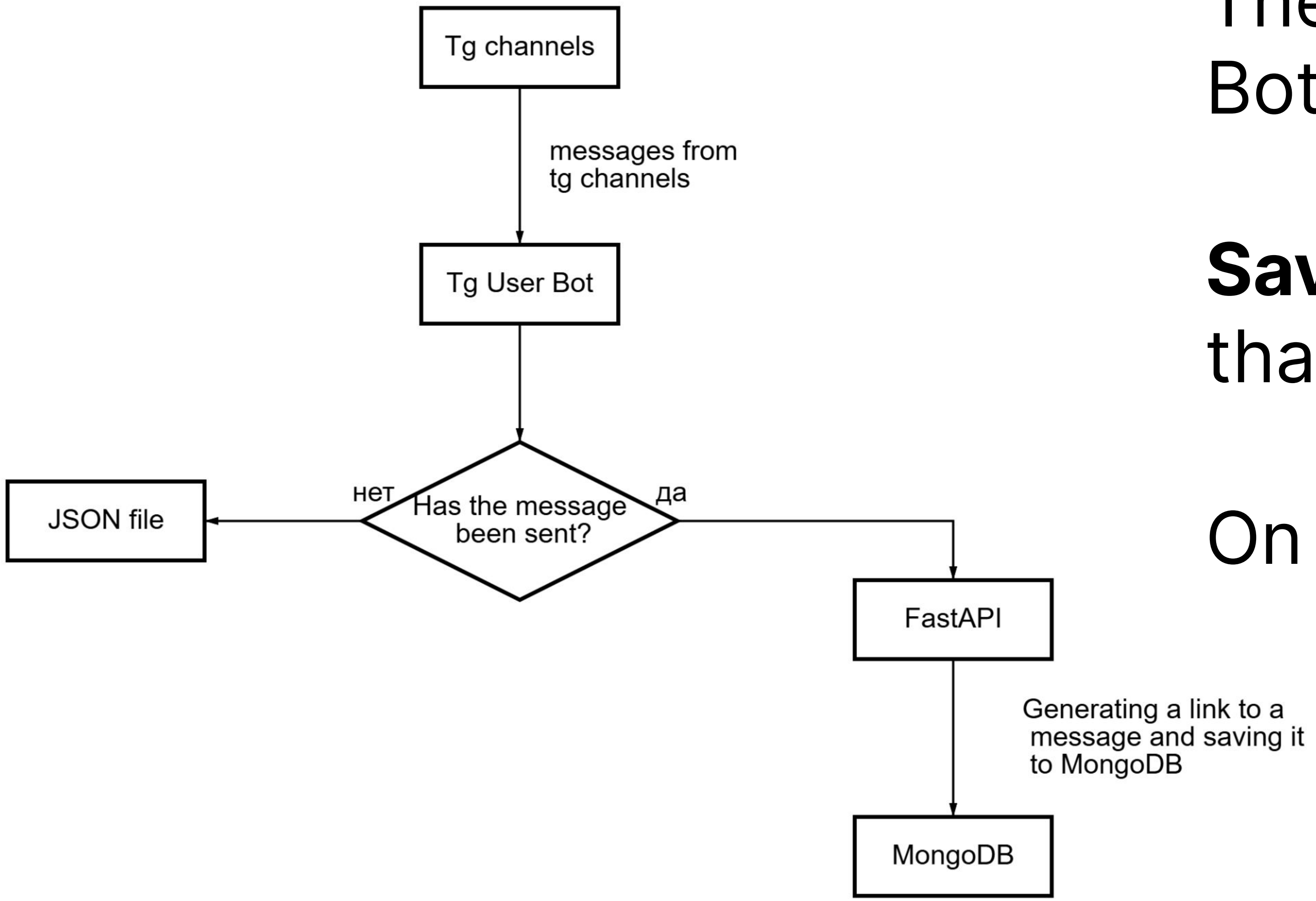
handles user’ search queries and redirects to ML worker

accepts Moodle courses from browser extension instances

accepts Telegram posts from User bot

perform document search based on file-metadata

telegram user bot


Tg channels
messages from tg channels
Tg User Bot
Has the message been sent?
нет
да
JSON file
FastAPI
Generating a link to a message and saving it to MongoDB
MongoDB

The library used for Telegram User Bot is **Pyrogram**

Saving and **resending** messages that failed to send previously

On **behalf** of the Telegram **user**

search by metadata

each Moodle file has the following metadata

```
class MoodleEntrySchema:
    course_fullname: str
    section_summary: str
    module_name: str
    filename: str
```

backend / stack

we use modern and popular technologies that allowed us to create a high-quality solution that can scale both horizontally and vertically

- Python 3.12
- Poetry - package and dependency management
- FastAPI - API implementation
- MinIO - local S3-storage for files
- MongoDB - scalable nosql database

solution / ml

accepts search requests from API

indexes all available corpora

performs semantic and syntax similarity search

runned on separate fast hardware

ml / stack

we used the tools that were most convenient for us

- Python 3.12
- Poetry - package and dependency management
- FastAPI - API implementation
- Qdrant - vector storage with batteries
- 🤗 Transformers - cozy library for ml models

ml / prepare

pdf files →

1. extract textual data
2. clean raw data
3. split into chunks based on tokenizer
4. saturate chunks with metadata
5. chunk → dense and sparse vectors
6. store chunk in Qdrant

- **ML components:**
  - Content Cleaner
  - Chunk generator
  - Hybrid Retriever (work on chunk level):
    - Dense Retriever:
      - Tokenizer
      - Embedding model
      - ANN Search & Builder (based on FAISS)
    - Sparse Retriever:
      - Pre-processing:
        - Tokenizer
        - Stemmer
        - Normalizer
      - BM25
    - Merger
    - Cross-Encoder
  - Metadata Retriever (work on file level):
    - TF-IDF based on MongoDB $text indexes
  - Rank Fusion (on file level)

ml / search

user text →

1. clean query
2. query → dense and sparse vectors
3. search 1000 by sparse, 1000 by dense in Qdrant
4. perform RRF fusion to get 100
5. rerank using CrossEncoder
6. collect chunks to documents

```
INFO:compute.search:Processing 1 items
INFO:compute.search:Text encoded by BiEncoder in 0.32 seconds
INFO:compute.search:Text encoded by BM25 in 0.00 seconds
INFO:compute.search:Qdrant Search completed in 0.12 seconds
INFO:compute.search:Reranked by CrossEncoder in 1.33 seconds
INFO:compute.search:Processed 1 tasks in 1.78 seconds
INFO:compute.search:Stored 2 results in 0.01 seconds
INFO:compute.search:Processing 1 items
```

ml / details

using Qdrant to store vector

- dense embeddings **stella_en_400M_v5** (current SOTA)
- sparse embeddings **BM25**
- **Reciprocal rank fusion** to merge ranking

and CrossEncoder **ms-marco-MiniLM-L-6-v2** as reranker

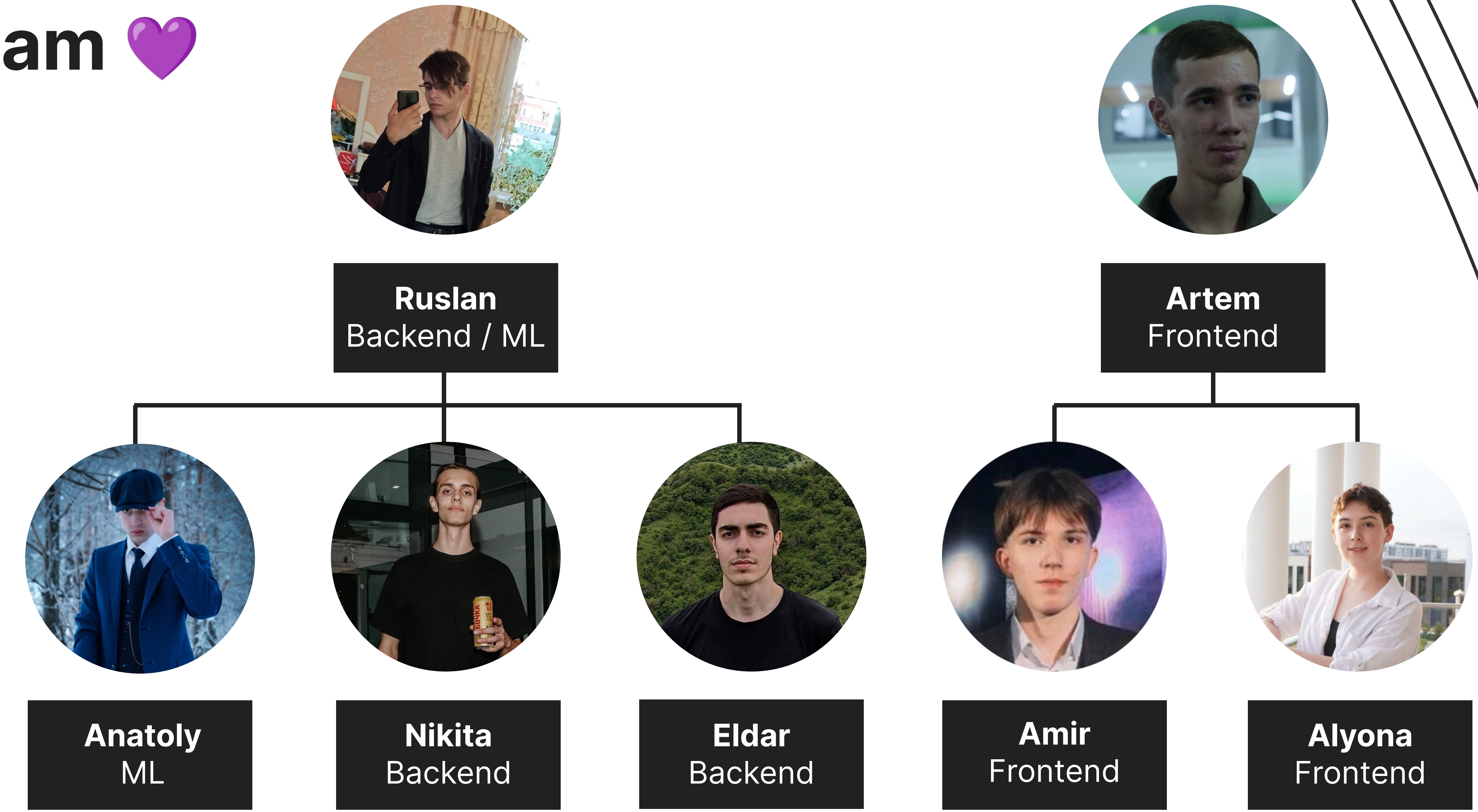
ml / experiments

we fix 1 hyperparam (model, preprocessing, etc.) and compare performance on the whole pipeline

Performance metrıcs:

0 rank awareness: Recall@10, Hit Rate@10
Rank awarened: MRR@10, NDCG@10

team 💜



Ruslan
Backend / ML

Artem
Frontend

Anatoly
ML

Nikita
Backend

Eldar
Backend

Amir
Frontend

Alyona
Frontend

plans

- add official documents scrapers
- index all telegram channels
- check students access rights
- ensure service stability before F24
- index all courses before F23

innohassle.ru / search