

Important Resources:

Repositories with extended documentation (for development and maintenance)

[Backend Repository](#) (extended information about environment variables etc.)

[Frontend Repository](#)

[Chat Service Repository](#)

[TypeScript Hybrid Types Module](#)

Service Structure:

[Draw.io Structure Map](#)

Service Directory Structure:

Each server/service has their port configured with apache (httpd) in the virtual machine and are accessible with

```
sudo nano /etc/httpd/conf.d/node.conf
```

```
sudo nano /etc/httpd/conf.d/https-node.conf
```

Backend:

```
va-backend/  
|  
├─ auth-server/  
├─ content-server/  
├─ upload-server/  
├─ sync-service/  
|  
├─ ecosystem.config.js  
├─ package.json  
├─ package-lock.json  
└─ README.md
```

Each server/service has its own .env file, and PM2 must always be started inside that server/service folder.

Each server/service is built with “npm run build” after changes and “pm2 restart all” will restart all existing PM2 Node processes.

Example pm2 startup command for auth-server (applies for all):

“pm2 start npm --name auth-server – start”

“pm2 logs” will show possible errors and logs.

Frontend:

```
va-frontend/
|
├─ src/                → Main application code (pages, components, styles)
├─ public/             → Static files (images, icons, etc.)
├─ node_modules/       → Installed dependencies
|
├─ next.config.ts      → Next.js configuration
├─ vite.config.ts      → Vite configuration
├─ leafletConfig.ts    → Leaflet map settings
├─ next-security.cjs   → Security settings for Next.js
|
├─ package.json        → Project info + scripts
├─ package-lock.json   → Dependency lock file
|
├─ tsconfig.json       → TypeScript configuration
├─ tsconfig.app.json   → Extended TS config
├─ next-env.d.ts       → Next.js TypeScript declarations
|
├─ eslint.config.mjs   → ESLint settings
├─ postcss.config.mjs  → PostCSS settings
├─ vitest-setup.js     → Test setup
|
├─ README.md          → Basic documentation
└─ script.sh           → Optional helper script
```

Frontend has its own environment variables as well in “.env”.

va-chat-service (OpenAI RAG Implementation):

```
va-chat-service/
|
├─ app/           → Main application code (API routes, UI if any)
├─ config/        → Configuration files for RAG, OpenAI, etc.
├─ stores/        → Data stores / vector store logic
├─ lib/           → Helper functions and shared utilities
├─ public/        → Static files
├─ node_modules/  → Installed dependencies
|
├─ ecosystem.config.js → PM2 configuration (run from this folder)
|
├─ package.json   → Project info + scripts
├─ package-lock.json → Dependency lock file
|
├─ next.config.mjs → Next.js configuration
├─ next-env.d.ts  → TypeScript declarations
├─ tsconfig.json  → TypeScript config
├─ eslint.config.mjs → ESLint rules
|
├─ README.md      → Documentation
└─ LICENSE        → License file
```

Has its own environment variables in the root and own PM2 process. Start-script is configured in package.json.

Ohjeet sovelluksen ylläpitäjille (Admin-ohjeistus vaihtokoordinaattoreille)

1. Yleiskuvaus

Sovellus tarjoaa Metropolian opiskelijoille ja henkilökunnalle helpon tavan löytää tietoa opiskelijavaihdosta, partnerikouluista ja vaihtoihin liittyvistä käytännöistä.

Käyttöliittymä on suunniteltu mobiili edellä, ja se sisältää interaktiivisia toiminnallisuuksia, kuten Leaflet-kartan ja tekoälyavustajan.

2. Tärkeimmät ominaisuudet

AI-keskusteluavustaja

- Käyttäjillä on käytössään reaaliaikainen AI-chat, josta he voivat kysyä neuvoja vaihtoihin liittyen.
- Ylläpidon ei tarvitse kouluttaa mallia itse; järjestelmä käyttää valmiiksi koulutettua avustajaa.
- Ylläpito on vastuussa tekoälyn käyttämisen drive kansion hallinnasta ja päivityksistä. Tekoäly oppii kansion sisältämistä tiedostoista.

Yhteydenottosivu

- Käyttäjät voivat ottaa yhteyttä ylläpitäjiin/koordinaattoreihin tarvittaessa.
- Ylläpitäjät voivat **muokata näkyvillä olevia yhteystietoja** (Nimi, sähköposti, titteli).
- Muista päivittää tiedot aina, kun organisaatiossa tapahtuu muutoksia.

Kohteet-sivu (Destinations)

- Sisältää ajantasaiset tiedot Metropolian vaihtokumppaneista.
- Interaktiivinen **Leaflet-kartta** näyttää korkeakoulut ja sijainnit.
- Admin voi päivittää:
 - Linkit joista kohteet haetaan, (ohjeet löytyy sivulta)

Responsiivinen käyttöliittymä

- Sovellus skaalautuu automaattisesti eri näyttöko'ille.
- Ei vaadi ylläpidolta toimenpiteitä, mutta sisällössä kannattaa suosia lyhyitä tekstikokonaisuuksia.

Google-kirjautuminen

- Käyttäjät voivat kirjautua Google-tilillään.
- Kirjautuminen on **rajoitettu Metropolian domainiin**, eli vain opiskelijat ja henkilöstö pääsevät sisään.
- Ylläpidon tehtävä:
 - Hallita tarvittaessa käyttöoikeuksia (esim. admin-roolien jakaminen).
 - Asiattomien käyttäjien poistaminen/blokkaaminen alustalta.

Käyttäjän tallennukset

Käyttäjät voivat tallentaa:

- omia tärkeitä dokumenttilinkkejään
- suosikkikohteita
- alustavan vaihtosuunnitelman (itinerary)
- taloussuunnitelman

Adminin ei tarvitse muokata näitä tietoja, ellei sitä erikseen pyydetä käyttäjältä tai ongelmatilanteessa.

Ajankohtainen vaihto-informaatio

- Käyttäjät näkevät ajantasaisen tiedon:

- hakuajoista
- tärkeistä päivämääristä
- prosessin vaiheista
- Admin vastaa näiden tietojen päivittämisestä tarvittaessa.

Taloudellinen tuki ja käytännön ohjeet

- Sisältää tietoa:
 - asumisesta
 - elinkustannuksista
 - opintotuesta
 - taloudellisesta avustuksesta
 - käytännön vinkeistä ennen vaihtoa ja vaihdon aikana
- Admin voi päivittää näitä ohjeita tarvittaessa.

Tärkeä huomio

- **Varsinaisia vaihtohakemuksia ei käsitellä tämän sovelluksen kautta.**
- Palvelu välittää vain tietoa ja ohjeita – hakuprosessi tapahtuu Metropolian virallisten kanavien kautta.