

ELO Learning

TECHNICAL INSTALLATION MANUAL V1

ZERO DAY

Last updated: 18 August 2025
University of Pretoria

Name	Student number
RM (Rene) Brancon	u22556771
NF (Nigel) Mofati	u22528084
TM (Tukelo) Mokwena	u22536800
S (Saskia) Steyn	u17267162
NG (Ntokoza) Tonga	u22506773

Team contact:

ZeroDay0D4y@gmail.com



Contents

1. Introduction.....	3
2. Prerequisites.....	3
3. Installation.....	3
3.1 Clone the Repositories.....	3
3.2 Backend Setup.....	4
3.3 Frontend Setup.....	4
4. Deployment / Running.....	5
4.1 Vercel Deployment.....	5
4.2 Render Deployment.....	5
4.3 CI/CD.....	5
4.3 Testing.....	5
5. Additional Notes.....	6
6. Screenshots.....	6
Backend:.....	6
Frontend:.....	6
Postman Test Example.....	7

1. Introduction

ELO Learning is a gamified mathematics learning platform designed to match learners based on their skill level using the ELO rating algorithm. The system is composed of:

- A **React + Next.js** frontend
- A **Node.js + Express** backend
- A **PostgreSQL** database
- A CI/CD pipeline using **GitHub Actions**

This document provides comprehensive setup instructions for developers or testers wishing to run the system locally, or access the deployed version at www.elo-learning.co.za.

2. Prerequisites

Ensure the following tools are installed on your system:

Tool	Version	Installation Guide
Node.js	>= 16.x	nodejs.org
Yarn (or npm)	Latest	npm install -g yarn
Git	Latest	git-scm.com
PostgreSQL	>= 13.x	postgresql.org
Docker (Optional)	Latest	docker.com
Supabase CLI (Optional)	Latest	supabase.com

3. Installation

3.1 Clone the Repositories

HTTP:

```
git clone https://github.com/COS301-SE-2025/ELO-Learning.git
cd ELO-Learning
```

SSH (recommended):

```
git clone git@github.com:COS301-SE-2025/ELO-Learning.git
cd ELO-Learning
```

3.2 Backend Setup

```
cd backend
# Create a .env file manually in the backend directory and add your DB
credentials, ports, and secrets

# Install dependencies
npm install

# Start development server
node src/server.js
```

3.3 Frontend Setup

```
cd frontend

# Create a .env file manually in the frontend directory and set
NEXT_PUBLIC_API_URL to your backend URL

# Install dependencies
yarn install
yarn build

# Start development server
yarn dev
```

The frontend should now be available at <http://localhost:8080> and the backend at <http://localhost:3000>.

4. Deployment / Running

4.1 Vercel Deployment

The application has been deployed and is accessible at: www.elo-learning.co.za.

(If re-deployment is required, please refer to the hosting provider's documentation or DevOps team instructions.)

4.2 Render Deployment

The services used by the application are deployed and accessible at: <https://elo-learning.onrender.com>

4.3 CI/CD

All commits to the **development** branch trigger GitHub Actions which:

- Run backend and frontend lint tests
- Execute Jest unit tests for backend
- Run Cypress integration tests for frontend
- Validate database schema sync

All commits to the **main** branch trigger GitHub Actions which:

- Run backend and frontend lint tests
- Execute Jest unit tests for backend
- Run Cypress integration tests for frontend
- Validate database schema sync
- Deploy to Vercel
- Deploy the backend to render

4.3 Testing

```
# Backend (Jest)
cd backend
npm test

# Frontend (Cypress)*
cd frontend
npx cypress open
```

Ensure Cypress is properly configured with the base URL pointing to your local or deployed frontend instance.

*For the Cypress tests to run, it is required that the frontend be running.

5. Additional Notes

- Use Supabase or pgAdmin to inspect or manually edit the PostgreSQL database.
- Use Postman to test REST API endpoints (/api/...).
- Refer to the [User Manual](#) for usage instructions after deployment.

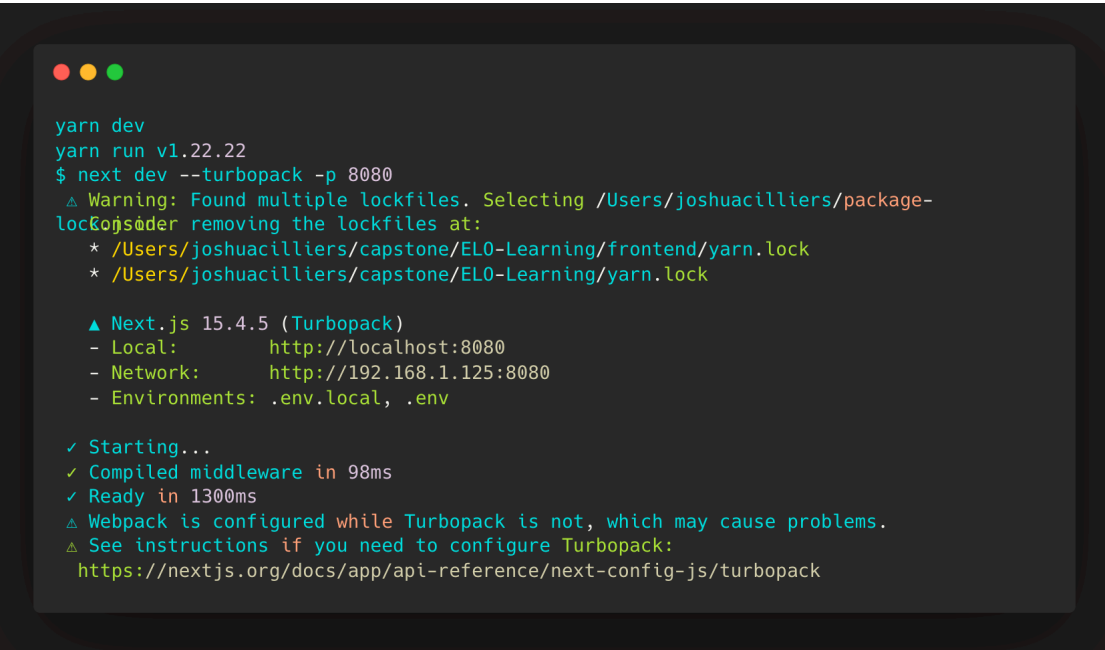
6. Screenshots

Backend:

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. The terminal text shows the command to run the backend server and the resulting output.

```
backend git:(main) node src/server.js
Server is running on
http://localhost:3000
```

Frontend:

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. The terminal text shows the command to run the frontend development server and the resulting output, including warnings and status messages.

```
yarn dev
yarn run v1.22.22
$ next dev --turbo pack -p 8080
  ▲ Warning: Found multiple lockfiles. Selecting /Users/joshuacilliers/package-
lock.json
  ▲ jsöder removing the lockfiles at:
    * /Users/joshuacilliers/capstone/EL0-Learning/frontend/yarn.lock
    * /Users/joshuacilliers/capstone/EL0-Learning/yarn.lock

  ▲ Next.js 15.4.5 (Turbo pack)
    - Local:      http://localhost:8080
    - Network:    http://192.168.1.125:8080
    - Environments: .env.local, .env

  ✓ Starting...
  ✓ Compiled middleware in 98ms
  ✓ Ready in 1300ms
  ▲ Webpack is configured while Turbo pack is not, which may cause problems.
  ▲ See instructions if you need to configure Turbo pack:
    https://nextjs.org/docs/app/api-reference/next-config-js/turbo pack
```

Postman Test Example

