

# Full Stack Developer

## Module: Final Project

## Requirements

### Disclaimer

---

- The information contained in this document is provided for informational purposes only.
- Although the CNFPC makes every effort to ensure that the information communicated is accurate and up to date.
- It declines all responsibility for any damage, loss or loss of profit, direct or indirect, caused by the use of such information.
- The contents of this document are for educational purposes only. They may be used by students enrolled in the relevant training for educational purposes only.
- All content is copyrighted, all rights reserved.

# 1 Project Overview

---

Build a full-stack application, demonstrating proficiency in both backend and frontend development, database design, API integration, and deployment.

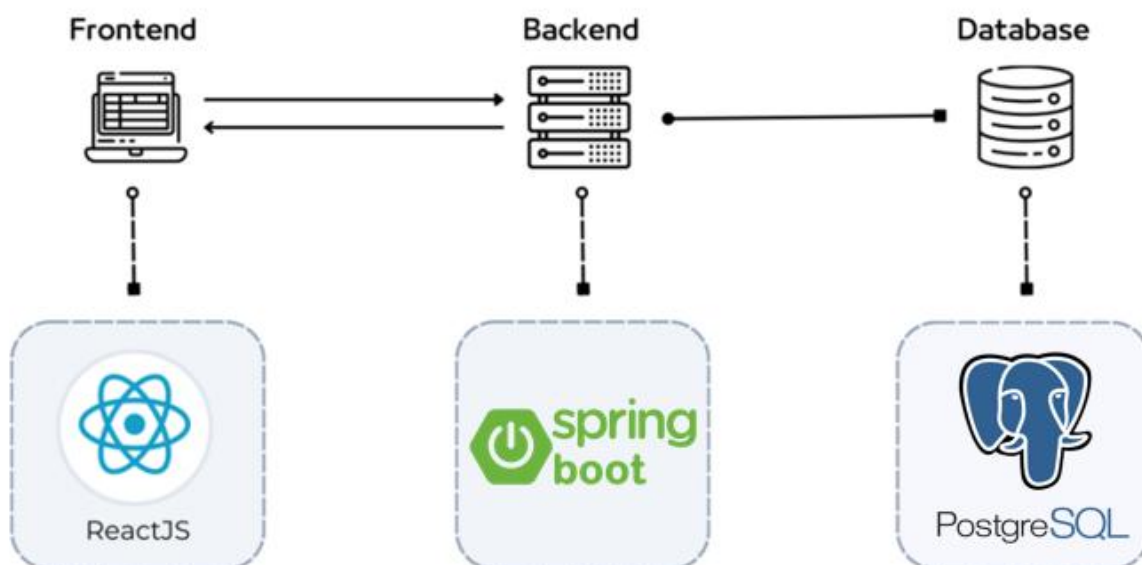
## 2 Source Control

---

- **Use Git for version control**
  - Use branches and git workflow by following best practices
  - Don't remove feature branches after merge (keep old branches)
  - Do several commits in a daily basis
  - Add teacher with "Reporter role" to your project.

## 3 Technical Requirements

---



## 3.1 Backend Development

### 3.1.1 Core Requirements

- Build RESTful API endpoints using Spring boot
- Implement proper error handling and data validation
- Follow Springboot best practices and conventions



### 3.1.2 Database Requirements

- PostgreSQL database implementation
- One One-to-Many relationship
- One Many-to-Many relationship



### 3.1.3 Authentication & Authorization

- Implement user authentication
- Secure password handling (hashed password in db...)
- Role-based access control

## 3.2 Frontend Development

### Frontend with ReactJs

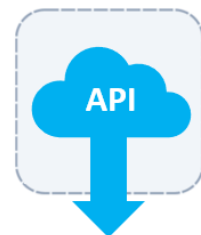
- Build a responsive design web app
- Implement proper state management
- Include form validation



**Note:** Frontend design should follow best practices on UX Design!

## 3.3 Optional: External API Integration

- Consume & integrate a **third-party API**, in order to use and process third-party data.



## 4 Project Deliverables

### 4.1.1 Source Code

- ☐ Well-organized (using branches) repository (GitLab or GitHub)
- ☐ README.md with setup instructions

### 4.1.2 Documentation

- ☐ Swagger API documentation
- ☐ Database schema

## 5 Project Individual Evaluation Criteria

### 5.1 Functionality (40%)

- ☐ All required features working
- ☐ Error handling

### 5.2 Code Quality (25%)

- ☐ Clean, well-organized code
- ☐ Proper commenting
- ☐ Best practices followed

### 5.3 Database Design (15%)

- ☐ Proper relationships
- ☐ Minimal table relationships requirements
- ☐ CRUD operations

### 5.4 User Interface (10%)

- ☐ Responsive design
- ☐ User experience

### 5.5 Documentation & Presentation (10%)

- ☐ Clarity
- ☐ Technical accuracy
- ☐ Git branches, commits, workflow

#### Grade Scale - Mention

- **54-60 points (90-100%):** Excellent
- **48-53 points (80-89%):** Very Good
- **42-47 points (70-79%):** Good
- **36-41 points (60-69%):** Satisfactory
- **Below 36 points (<60%):** Fail

## 6 Suggested Project Topics (Examples)

---

**Note:** Due to the 5-day timeline, focus on Simplicity. Do not try to build a real-world commercial application. Build a functional prototype that demonstrates CRUD operations and data flow.

### Travel Wishlist (Simplified Travel Agency)

- **Scope:** build a "Travel Wishlist" manager.
- **Core Features:**
  - **Admin:** Create "Destinations".
  - **User:** Browse destinations and add them to a "My Wishlist".
- **Optional AI Integration:** A "Suggest Me" button. User enters "Beach, Cheap", and the AI returns a text suggestion of a country.

### Event Board (Simplified Event Management)

- **Scope:** A simple bulletin board for company events.
- **Core Features:**
  - **Admin:** Post new events (Date, Title, Location).
  - **User:** Click "Attend" on an event (adds user to the participant list).

### Office Gear Rental (Simplified Rental)

- **Scope:** Manage borrowing internal office equipment (laptops, adapters)
- **Core Features:**
  - **Inventory:** List items with a status (Available / Borrowed).
  - **Action:** User clicks "Borrow" -> Item status changes to "Borrowed" and links to that User.

### IT Service Desk

- **Scope:** A simple ticketing system for reporting bugs.
- **Core Features:**
  - **User:** Create a Ticket (Subject, Description, Urgency).
  - **Admin:** View list of tickets and change status (Open -> Solved).
- **Optional AI Integration:** "Sentiment Detector." AI analyses the ticket text. If the user sounds angry, it automatically flags the urgency as "High" or a sentiment score.

### Custom Topic

- Any other topic is allowed but must be validated by the trainer before starting.

## 7 Key dates

---

- **Tuesday, 20/01 or Wednesday, 21/01/2026:**
  - Progress Review (1-on-1 with Trainer)
- **Thursday, 22/01/2026**
  - Submission Deadline : by end of day
- **Friday, 23/01/2026:**
  - Final Project Assessment (30 mins – By appointment)

## 8 Submission Instructions

---

Submit your work by end of **Thursday the 22/01/2026**.

- Ensure Github or Gitlab repository is public OR share with read access to:
  - André Silva (GitHub: **souan**)
  - Nelson Ferreira (GitHub: **nels-ferr**)
- Submit repository link via [email or teams]

## 9 Questions?

---

See André Silva ([andre.silva@education.lu](mailto:andre.silva@education.lu)) or Nelson Ferreira ([nelson.ferreira@education.lu](mailto:nelson.ferreira@education.lu)) during class hours.

**Good luck with your projects!**