Project tools - considerations & notes

Java 17 choice:
https://sdkman.io/

- Apparently it's the version with most improvements since popular versions 8 & 11
- It has LTS (Long Term Support) label
- The most recent version (Java 21) was released in Sep 2023, and therefore may be considered the most stable, or "tried & true" to use for new production projects
- Java 17 Temurin distribution is apparently well supported (YouTube opinion)
- Java 17 is minimum version for a Spring Boot 3 application

IntelliJ IDE choice:
https://www.jetbrains.com/idea/

- I'm using the 30-day free trial of the full version (Ultimate edition), but it is also possible to install the Community version for free (has most features but apparently not all Spring components or support)

https://start.spring.io/ - website tool which enables you to initialise a new Spring Boot project using settings/dependencies you select

Spring Boot 3.2.0 choice:

- It's the current GA (generally available) release aka latest stable version
- Chose to build with Maven since it had a lot of support online
- Added these dependencies:
  - Spring Web as it comes with embedded web server Apache Tomcat - so the app can be put in production easily. Also gives access to Spring MVC (another framework within Spring) - this is the design pattern I want to use.
  - H2 database as it's an in-memory DB which is fully compatible with Spring Boot without need for configurations. However, H2 is unsupported for actual production usage so would likely use another DB in a real case.
  - Spring Data JPA as it allows for mapping Java objects to tables in a relational DB without need to use specific database semantics
  - Spring Actuator as it provides all of Spring Boot's production-ready features (as per the Spring online docs)
  - Spring Boot DevTools because it seemed useful
  - Spring Security for security tools for authentication and authorisation, etc. Decided not to enable OAuth2 client logins (google, twitter, etc) for the scope of this project.
  - Spring Session for managing user session data

- Decided not to use Lombok library for code generation as online discussions say it can often lead to hard-to-find bugs with JPA. Decided to use IDE code generation tools instead (for generating getters, setters, etc) and for ease of readability and adaptation as code scales.

Node.js 20.10.0 choice (includes npm 10.2.3):

- It's the latest LTS version so stable for building with
- Node.js is required for using React
- Additionally added tools using npm: Bootstrap, React Router, ReactStrap

Others choices:

- Using React 'proxy' for scope of this project; however, not suitable for production if I had a real live domain. Would need to use something like Express or Nginx.

- Currently I'm throwing more top-level generic exceptions like Exception and RuntimeException (with specific messages) as opposed to specific subsets like IllegalArgumentException as it was quicker for the scope of this project, and is still effective.