






# Tech stack choices




Recommended IDE: vscode <https://code.visualstudio.com/>

Version Control Tool:  &  [GitHub: Let's build from here](#)


## Front-End: Next.js

- focus on Next.js's file system routing  [Building Your Application: Routing](#)
  - Not gonna consider Next.js's server-side rendering (SSR) feature - i.e. all our components would be client component
- in TypeScript
  - Styling:
    - TailwindCSS  [Tailwind CSS - Rapidly build modern websites without ever leaving your HTML.](#)
  - Request-response: axios  [Getting Started | Axios Docs](#)
  - Form submission: react-hook-form  [Home](#)
  - drag and drop (for re-ordering questions when creating a test):  [GitHub - atlassian/react-beautiful-dnd: Beautiful and accessible drag and drop for lists with React](#)
  - Global state management: react context API
  - UI lib (e.g. mui)

## Back-End: Nest.js

- API: RESTful APIs
- In TypeScript
  - JWT token for authentication & authorization
- API document: swagger  [Documentation | NestJS - A progressive Node.js framework](#)
  - since postman team only supports 3 team member, we need to make our API document available to the other team
- learning materials:
  -  [Documentation | NestJS - A progressive Node.js framework](#)
  -  [NestJS: The Complete Developer's Guide](#)

## Database

- MongoDB Atlas
  - Straightforward deployment as it is out of box
  - Use 3rd party lib - "Mongoose"  [Mongoose v8.3.2: Getting Started](#) at the backend to access MongoDB