FINLEARN

AGILE
AVENGERS

FINAL DEMO

TABLE OF CONTENTS

1. Background
2. Demo
3. Process
4. Software Architecture
5. Technical

FINLEARN

WHAT ARE WE TRYING TO SOLVE?

IMPROVE FINANCIAL LITERACY AMONG YOUNG ADULTS

Lack of Financial Literacy:

- Complex terms and concepts in finance
- Too many sources of information
- Inaccurate and conflicting advice

Fear of making mistakes:

- Avoid risk even favourable
- Avoid investing
- Uninformed decisions due to peer pressure

Disengaging traditional education:

- Textbooks are dull
- All theoretical learnings are eventually forgotten if not applied
- Very few avenues of liability-free practical application

FINLEARN

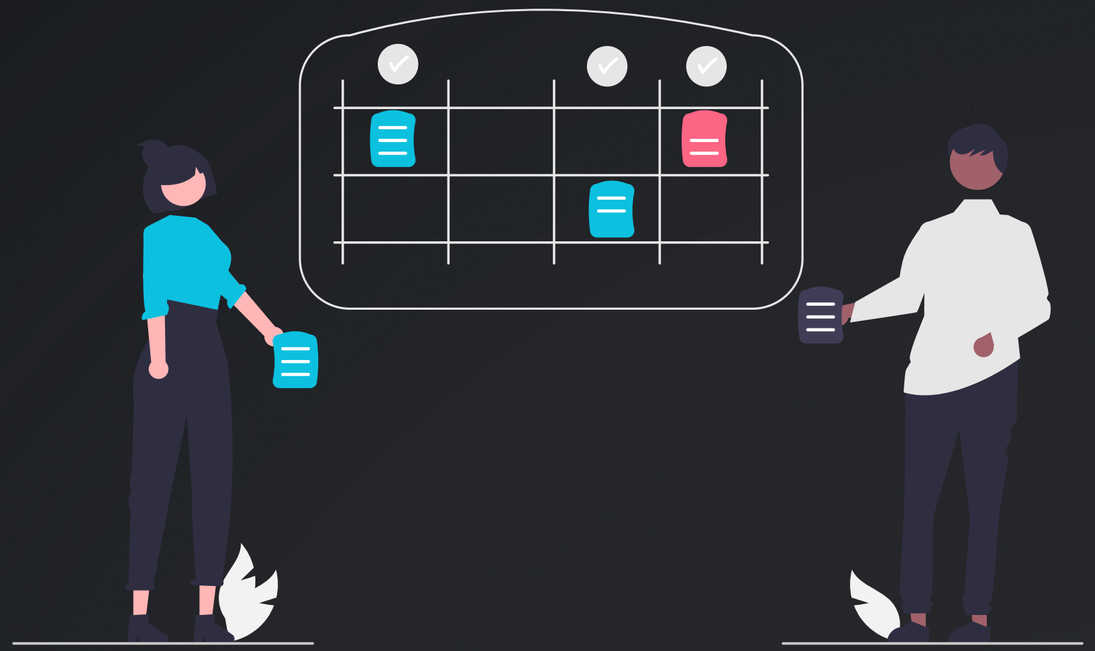
OUR SOLUTION

FINLEARN

FinLearn is
- a financial literacy platform
- for young adults
- who are ready to take a step towards an independent lifestyle.

FinLearn offers
- an all-in-one
- zero-liability
- learning solution

FinLearn does this by providing
- short lessons in the form of videos and articles
- a sandbox environment to test theoretical learnings



FINLEARN

PROCESS

A procedure for how we worked to accomplish this task

We conducted daily stand-ups along with sprint reviews and sprint planning meetings once a sprint.

FINLEARN

EXPRESS AND IDEATE **1**

Communicate needs and tasks using Discord, Zoom, and in-person discussions. Document decisions on Confluence.

DEFINE AND ASSIGN **2**

All stories and tasks were logged on JIRA and assigned to members based on their knowledge and interest.

WORK AND REVIEW **3**

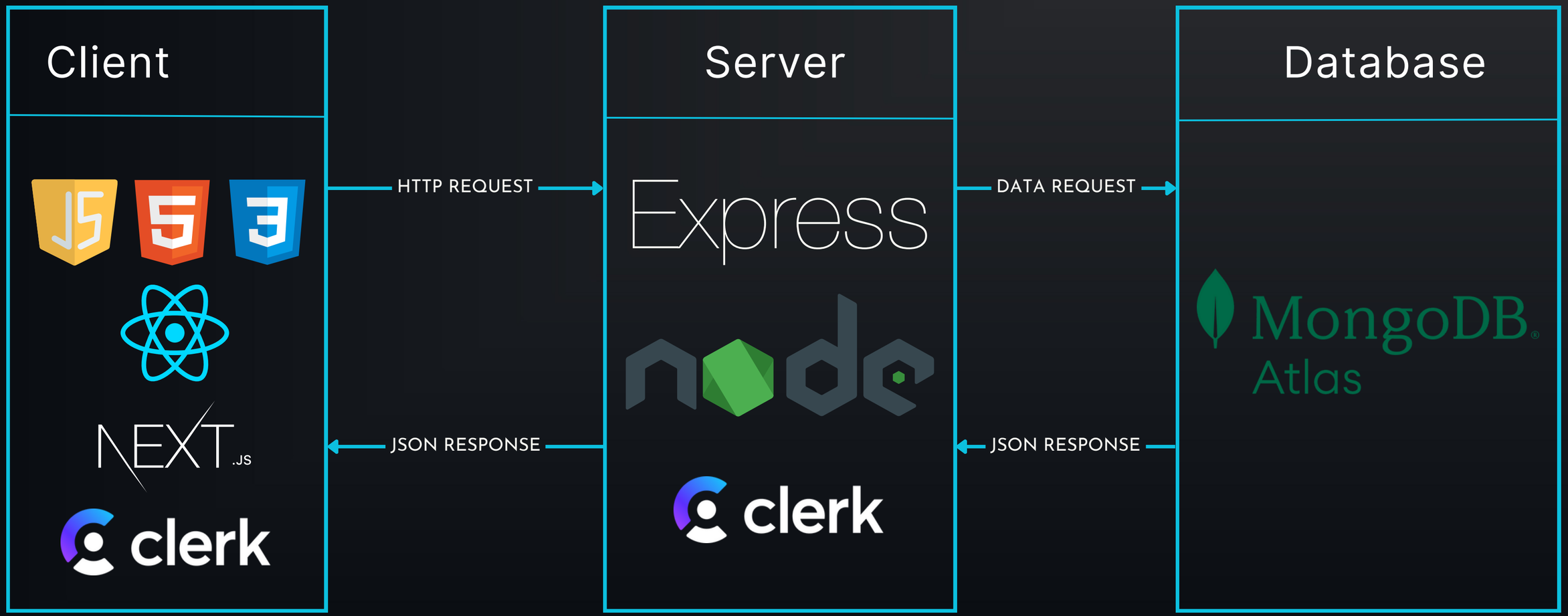
We used Git for version control and GitHub for source control and reviews.

PRS AND COMPLETION **4**

After a feature met the acceptance criteria, the branch was merged to develop and the ticket was moved to Done. All features in the sprint were merged at the end to the main.

SOFTWARE ARCHITECHTURE

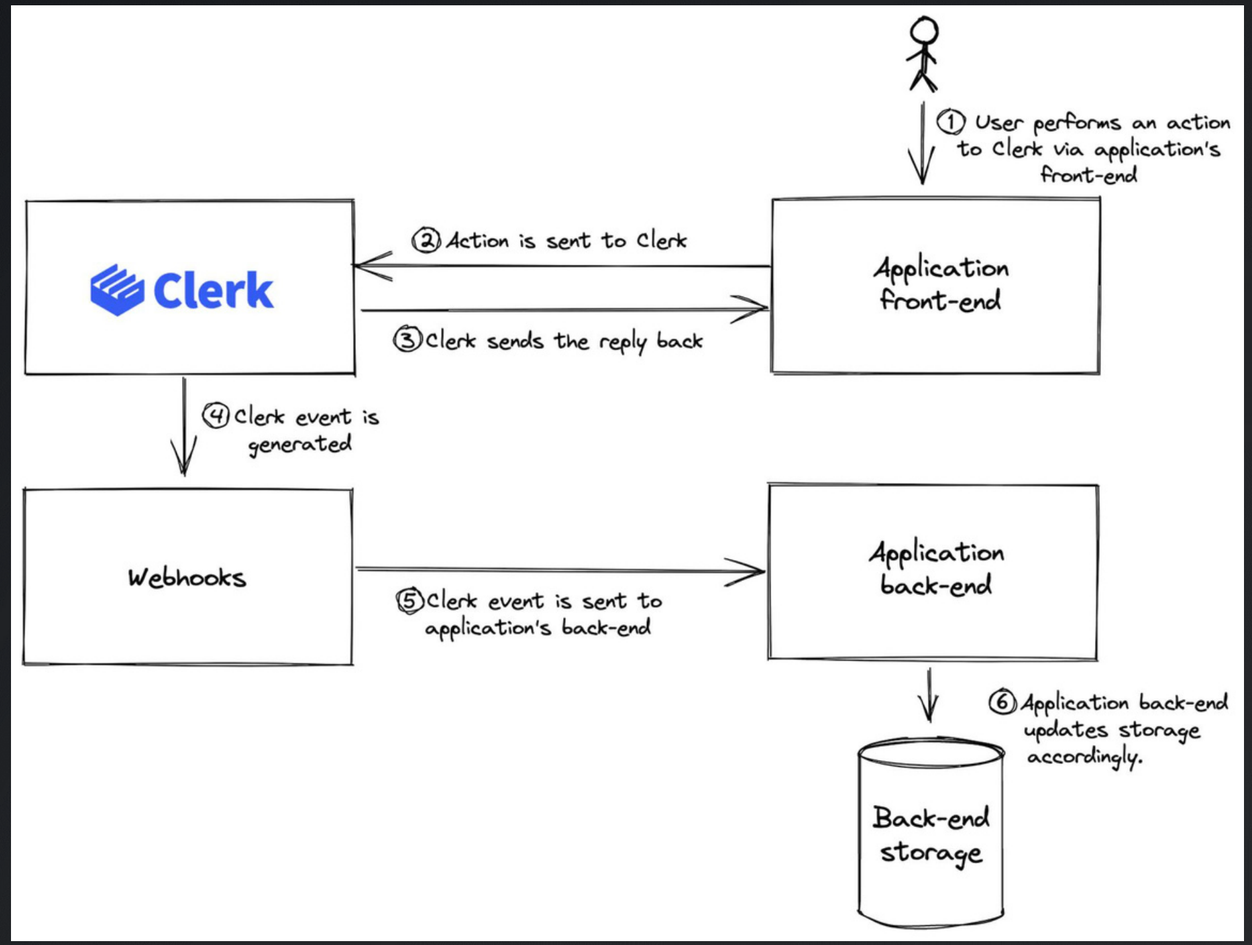
FINLEARN



TECHNICAL

Webhooks

- To keep our database in sync with users being registered on Clerk, we make use of Webhook
- Webhooks allow Clerk to inform us about specific events



FINLEARN

TECHNICAL

Video Progress Tracking

- Firstly, our initial choice of the learning progress schema closely resembled an entity-relationship schema commonly used in SQL databases. This led to a need for a complex populating logic
- On the front-end we needed a timer to track, the video's progress. This timer was designed to trigger a specific action every 5 seconds of active video watching by a user.

Example Object Used to track Learning Progress

FINLEARN

INDIVIDUAL CONTRIBUTIONS

- Adam
- Aditya
- Aryan
- Balraj
- Rahul
- Shashwat
- Shawn

FINLEARN