QUIZCRAFT

Tarik Spahic, Tamara Puzić, Tobias Fuchs

Group 2

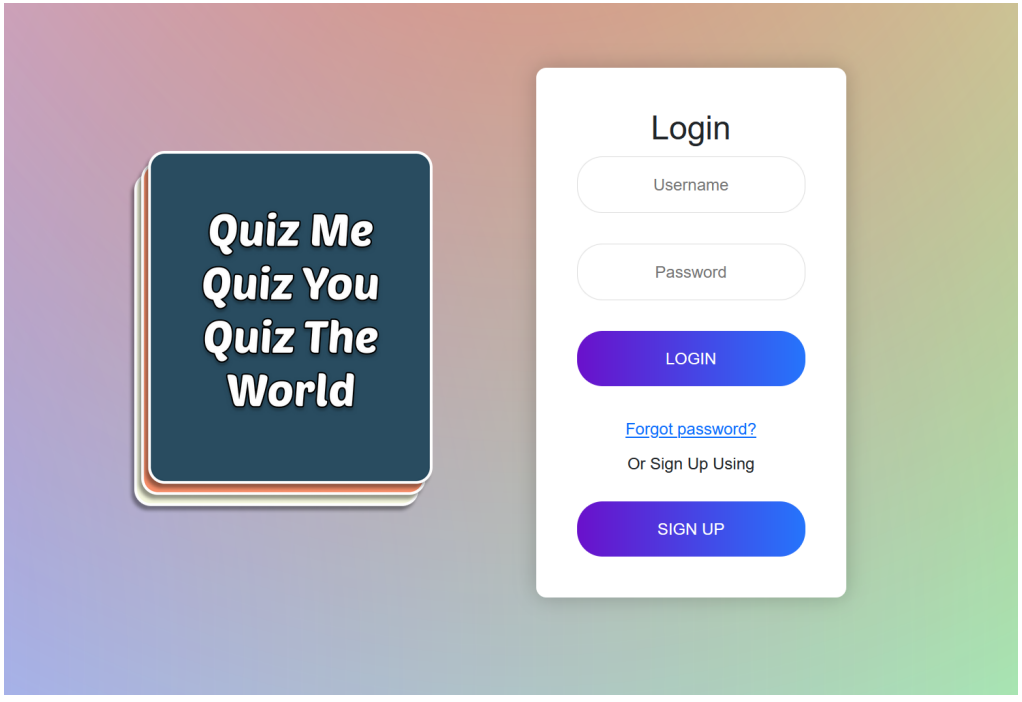
Agile Software Development SS24

Outline

Outline

Introduction - QUIZCRAFT

Test your knowledge!



Introduction - Project Idea

- Make a single-window trivia-quiz browser game
- Support multiplayer and user registration
- Allow user to customise his/her experience for a game
- Save stats for each user onto the database
- Have a unique and competitive experience in each game mode

Outline

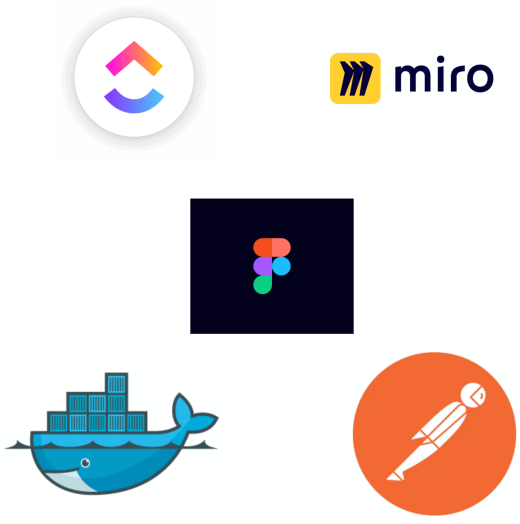
Organisational Aspects - Agile Methods

Why and how did we use them?

- All of the tasks and milestones were split into sprints
- Each sprint had a certain common goal to be reached
- Creating tickets and dividing them as-equal-as-possible
- Bigger tickets were split into smaller
- Multiple meetings per week
- Pair programming sessions for specific features

Organisational Aspects - Tools

- Brainstorming = Miro
- Prototyping = Figma
- Sprint and Tickets = ClickUp
- Testing APIs = Postman
- Deployment = Gitlab CI/Docker



Outline

Technical Aspects - Architecture

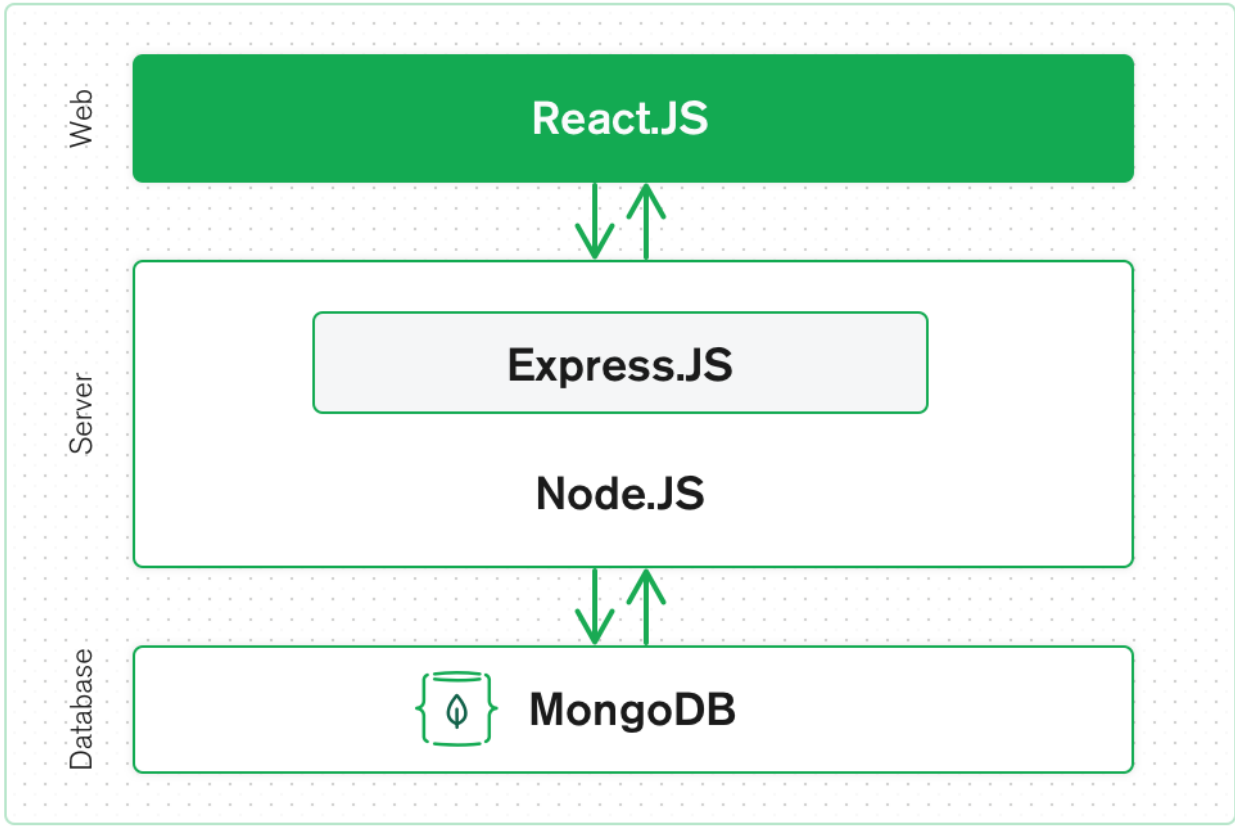


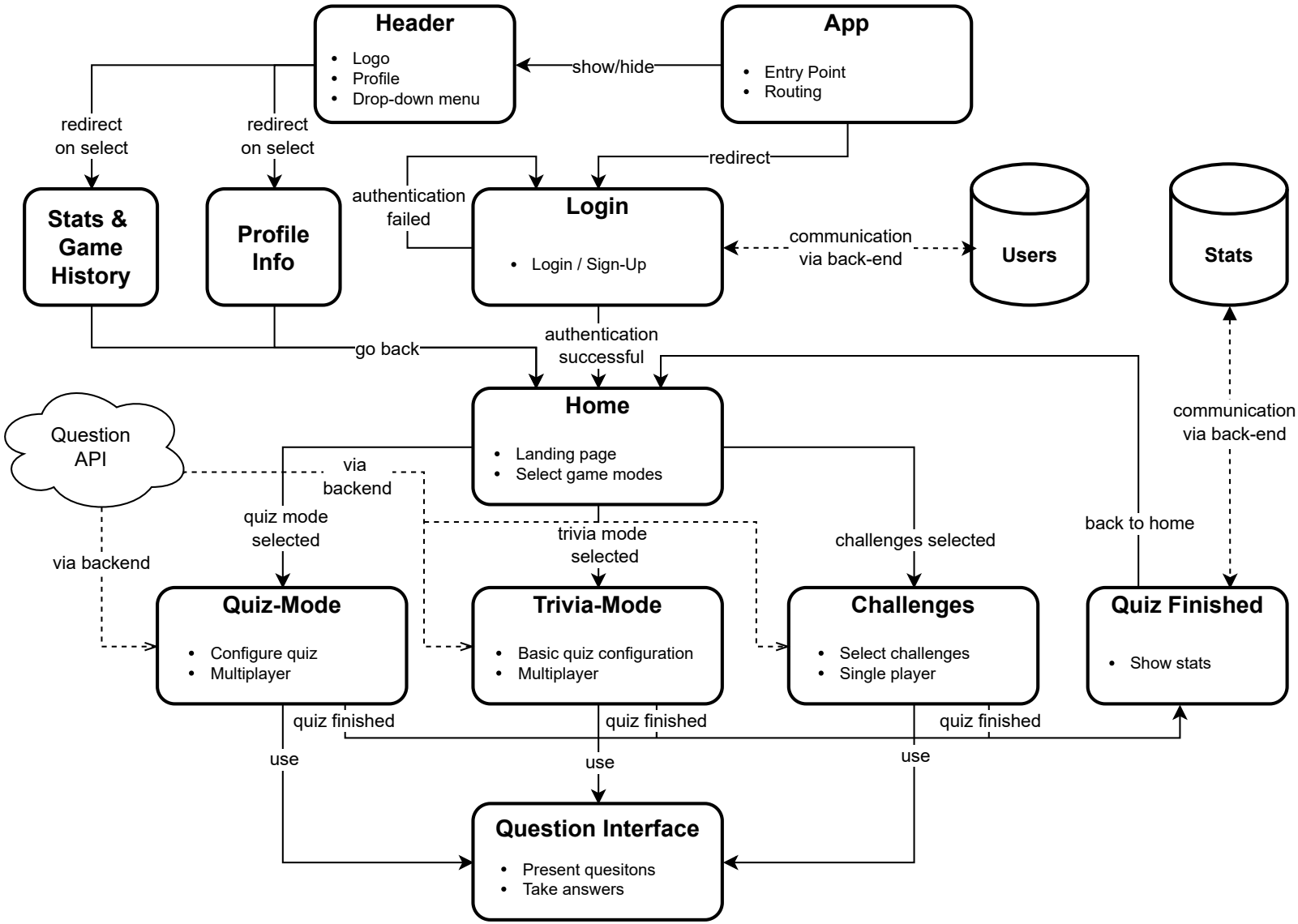
Figure: MERN tech stack (source: https://www.mongodb.com/resources/languages/mern-stack)

Technical Aspects - Front-end

React.js

- Component based UI architecture: components can be re-used
- Bootstrap for basic styles
- Communication with the back-end via web socket events

Technical Aspects - Front-end Component Hierarcy



Technical Aspects - Back-end

Node.js

- Game/session management: handles multiple clients at a time
- Supports single and 1v1 multiplayer games using socket rooms
- Question fetching via API
- Database communication
- User authentication

MongoDB

- Store user information
- Store game stats

Technical Aspects - Authentication

User authentication in our system involves several key steps:

User Registration

- Username, email, and password input
- Password must match special criteria for security

Password Storage

- Passwords hashed using bcrypt, no plaintext is stored

Login Process

- Username and password submission
- Password hashing and comparison
- User authentication and session initiation

Password Reset

- Email-based password reset
- Temporary password generation and hashing

These steps ensure secure handling of user credentials and authentication.

Technical Aspects - Database

MongoDB

- Stores user credentials and details
- Manages user sessions and activities
- Stores game statistics and player performance data

Database Communication

- Uses Mongoose ODM for interaction
- Establishes connection using connection URI
- CRUD operations for user and game data management

These practices ensure reliable data storage and retrieval.

Technical Aspects - Question API

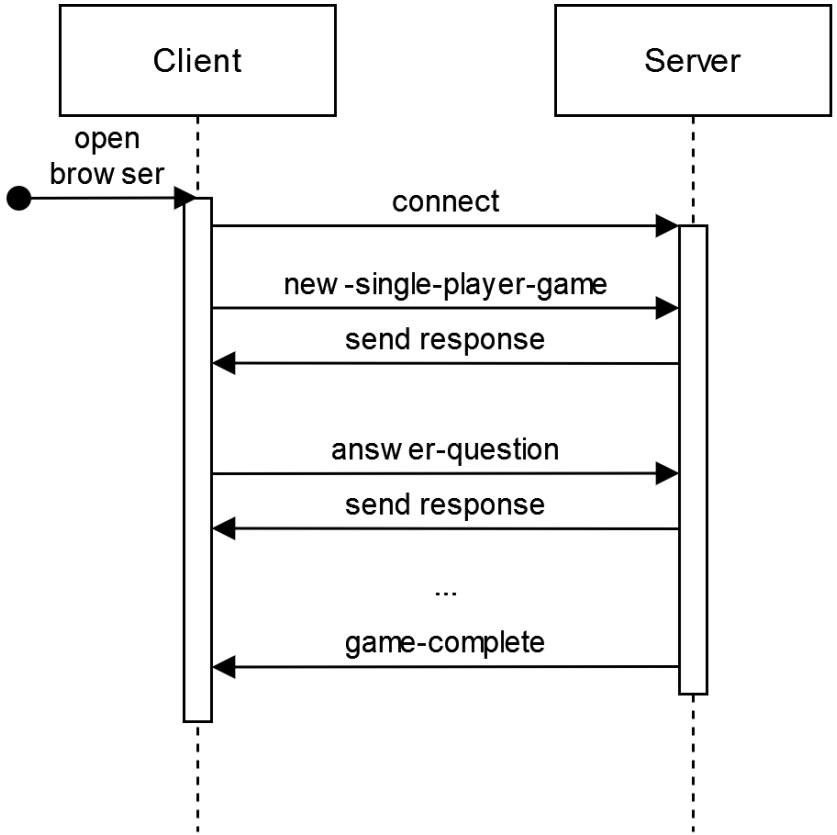
Open Trivia API

- Provides arbitrary questions
- Can specify question type, category, difficulty
- Use session tokens to get no duplicate questions
- Example: `https://opentdb.com/api.php?amount=10&category=21&difficulty=easy&type=multiple`
- **IP restriction** – only one server API request every 5 seconds
  - How to handle error responses?
  - How to handle multiple request calls within 5 seconds?
    $\rightarrow$ Keep track of latest API call and wait for 5 seconds before making another one

Technical Aspects - Client-Server Communication

Example: Single-player game

- Clients connects to server
- Both clients and server emit and listen to events at all times
- Server notifies clients upon game end
- Bidirectional communication via socket.io



Technical Aspects - Multiplayer

How to handle multiplayer games?

- Clients join multiplayer queue
  - Match clients with identical game settings
  - Clients need to wait for match $\rightarrow$ show waiting screen
  - Join multiplayer socket room for mutual communication
- Start multiplayer game at the same time (2 players)
- Server waits for both clients to answer questions
  - The client to answer first needs to wait for the other client
  - Once all clients answered, the server sends a response
- Analogue to single-player, all clients are notified upon game end

Technical Aspects - Room for Improvement

- **Security:** password storage, authentication procedure
- **Scalability:** game management, session handling
- **Performance:** question fetching
- **Usability:** question quality/diversity, quiz configurability
- **Stats Tracking:** granularity, user display, leader-board(s)
- **Multiplayer:** inviting friends / hosting games, multiplayer challenges
- **Hosting:** the back-end server is not hosted
- **Testing:** only very limited testing, manual testing

Outline