Writing a full stack application

Dr. Peter Dillinger
peter.dillinger@email.de

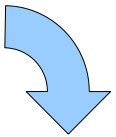
Content

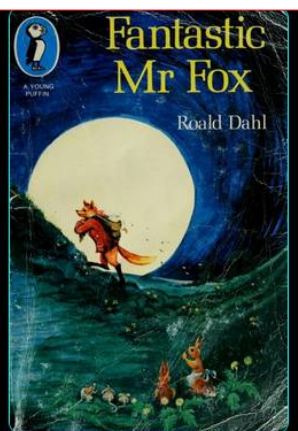
- Defining the application goal and architecture
- Development steps
  - Setting up the database
  - Implementing the backend and the frontend
  - Testing
  - Deploying
- Summary
- Literature and Sources
- Quiz

Additional information and material
https://github.com/phd4hd/SRH-HS-HD

```
_id: ObjectId('646bb1e2acab25490ddfea0e')
title: "The Da Vinci Code"
authors: Array
isbn10: "0385504209"
year: "2003"
cover: "https://covers.openlibrary.org/b/id/12521602-L.jpg"
reviewIds: Array

_id: ObjectId('646bb1e2acab25490ddfea0f')
title: "Fantastic Mr. Fox"
authors: Array
  0: "Roald Dahl"
  1: "Quentin Blake"
isbn10: "0140306765"
year: "1975"
cover: "https://covers.openlibrary.org/b/id/12374431-L.jpg"
reviewIds: Array

_id: ObjectId('646bb1e2acab25490ddfea10')
title: "The Kite Runner"
authors: Array
```


Database

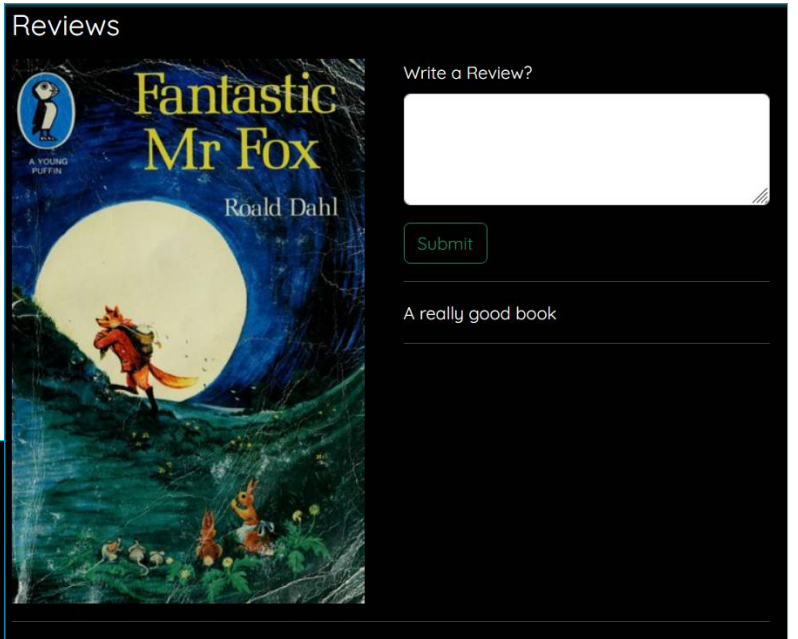


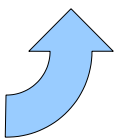


Roald Dahl and
Quentin Blake

Fantastic Mr.
Fox

Reviews

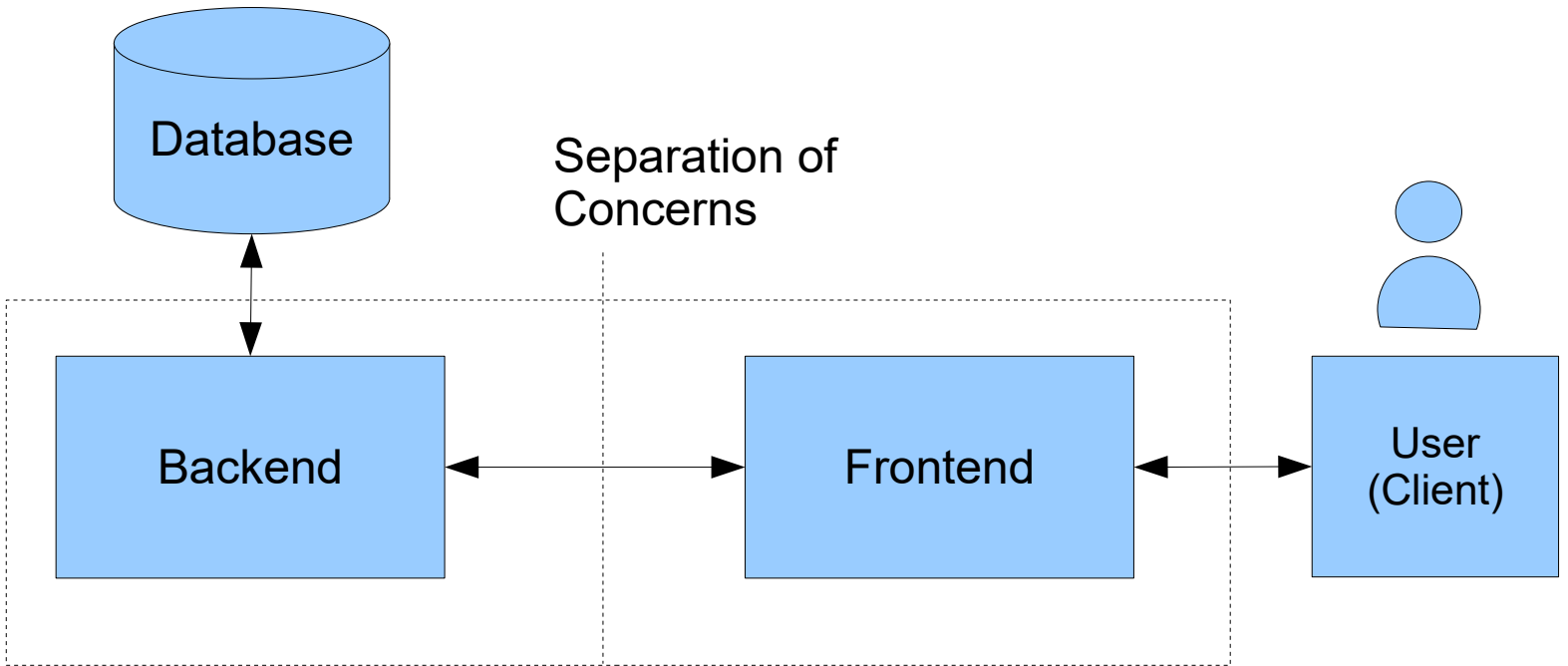




Reviews

Overview

Architecture



Full stack (more than one layer)

Data Management

fullstack_db.books


```
_id: ObjectId('646bb1e2acab25490ddfea0e')
title: "The Da Vinci Code"
authors: Array
isbn10: "0385504209"
year: "2003"
cover: "https://covers.openlibrary.org/b/id/12521602-L.jpg"
reviewIds: Array

_id: ObjectId('646bb1e2acab25490ddfea0f')
title: "Fantastic Mr. Fox"
authors: Array
  0: "Roald Dahl"
  1: "Quentin Blake"
isbn10: "0140306765"
year: "1975"
cover: "https://covers.openlibrary.org/b/id/12374431-L.jpg"
reviewIds: Array

_id: ObjectId('646bb1e2acab25490ddfea10')
title: "The Kite Runner"
authors: Array
```


```java
@Document(collection = "books")
@Data
@AllArgsConstructor
@NoArgsConstructor
public class Book {
    @Id
    private ObjectId id;

    private String title;
    private List<String> authors;
    private String isbn10;
    private String year;
    private String cover;
    @DocumentReference
    private List<Review> reviewIds;
}
```

Book.java

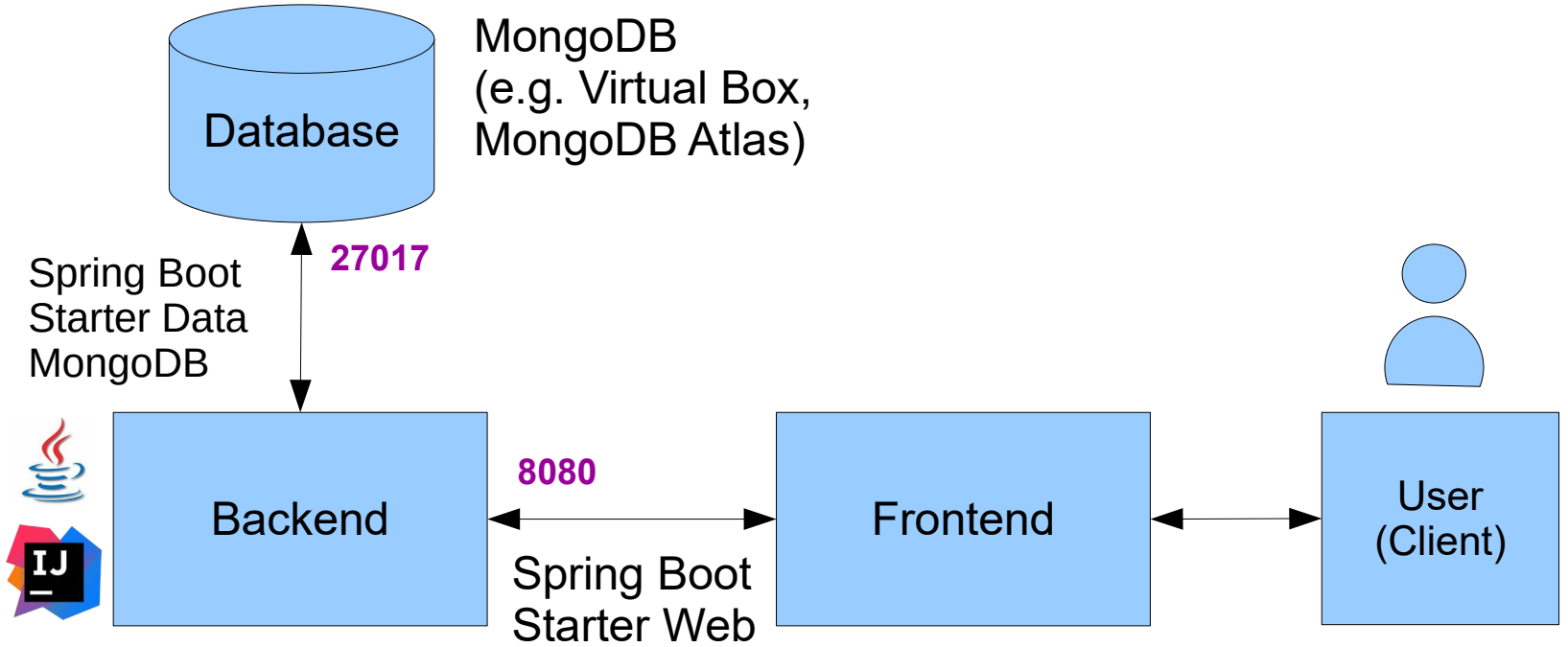
Review.java

```java
@Document(collection = "reviews")
@Data
@AllArgsConstructor
@NoArgsConstructor
public class Review {
    @Id
    private ObjectId id;
    private String text;

    public Review(String text) {
        this.text = text;
    }
}
```

Spring Boot Starter Data MongoDB

Backend

fullstack_db




```
GET   /api/v1/books
GET   /api/v1/books/{isbn10}
GET   /api/v1/reviews
POST /api/v1/reviews, {isbn10,reviewText}
```

Backend

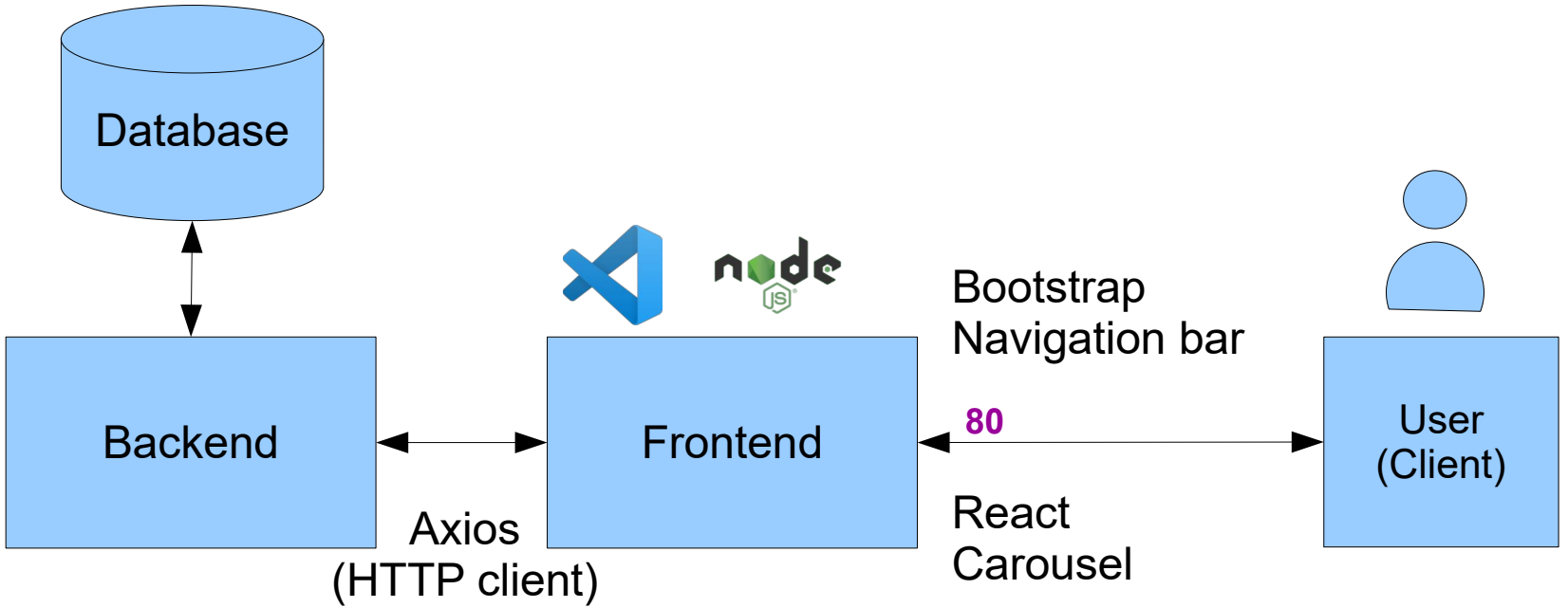
```java
@RestController
@RequestMapping("/api/v1/reviews")
public class ReviewController {
    @Autowired
    private ReviewService service;

    @GetMapping
    public ResponseEntity<List<Review>> getReviews() {
        return new ResponseEntity<>(
                service.findAllReviews(), HttpStatus.OK);
    }

    @PostMapping()
    public ResponseEntity<Review> createReview(@RequestBody Map<String, String> payload) {
        return new ResponseEntity<Review>(
                service.createReview(
                        payload.get("isbn10"),
                        payload.get("reviewText")
                ), HttpStatus.OK);
    }
}
```

REST endpoint (resource)

GET request on this resource will return all reviews of all books

POST request on this resource will create a new entry in the review collection

Frontend


Database
Backend
Axios
(HTTP client)
Frontend
Bootstrap
Navigation bar
80
React
Carousel
User
(Client)

Frontend (with light theme)

Bootstrap Navigation bar

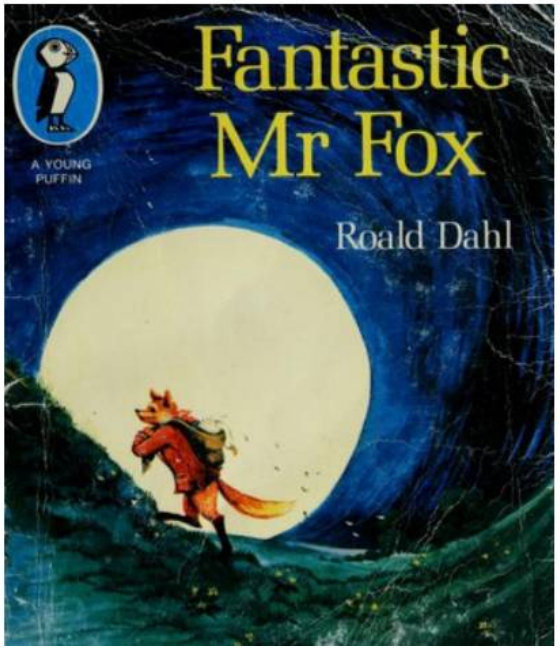
HTTP Navigation

div: book-card-container

React Carousel

Frontend (with light theme)

Reviews



React ReviewForm

Write a Review?

Submit

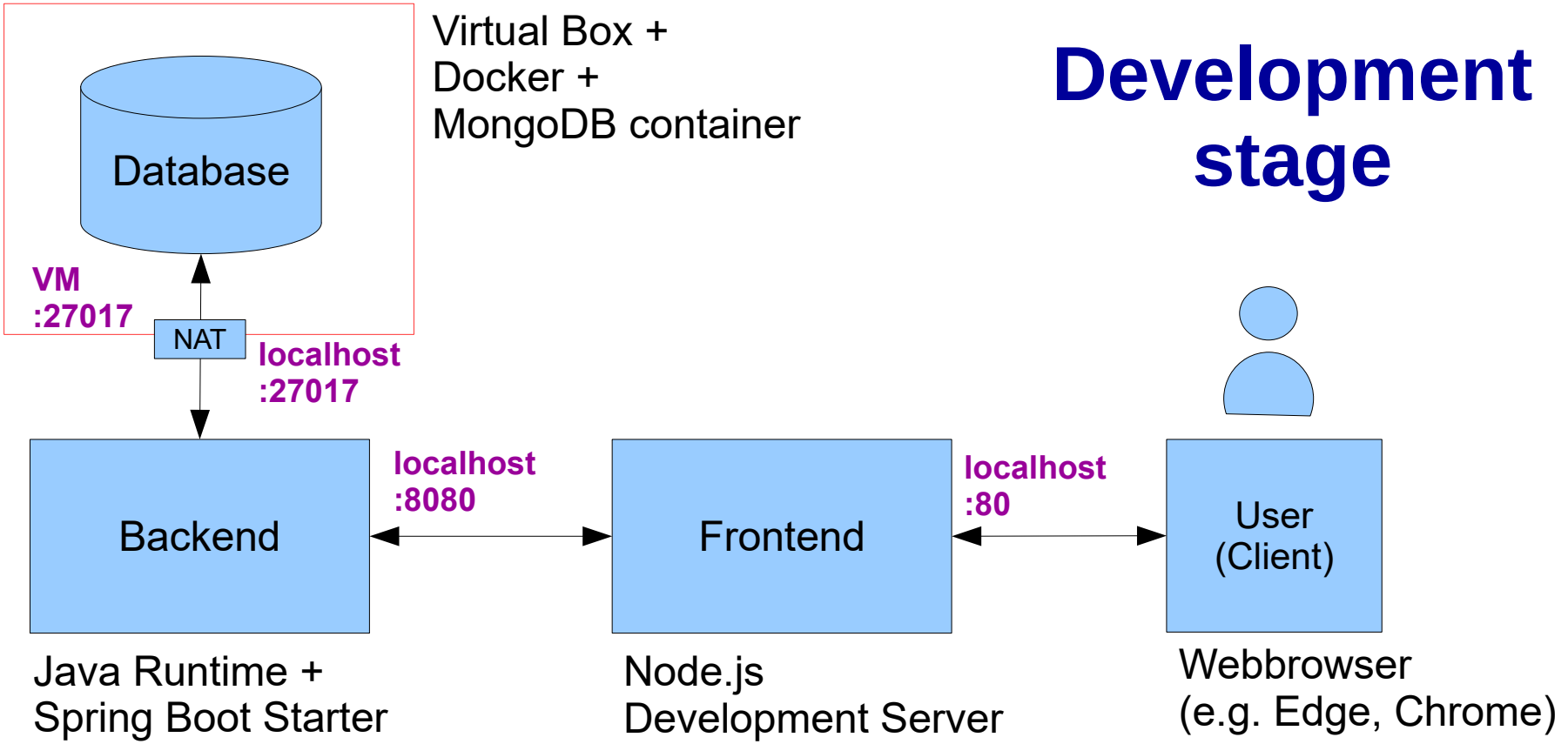
Really good book!

Is it working now?

Nice :)

Row/Col Table

Development stage



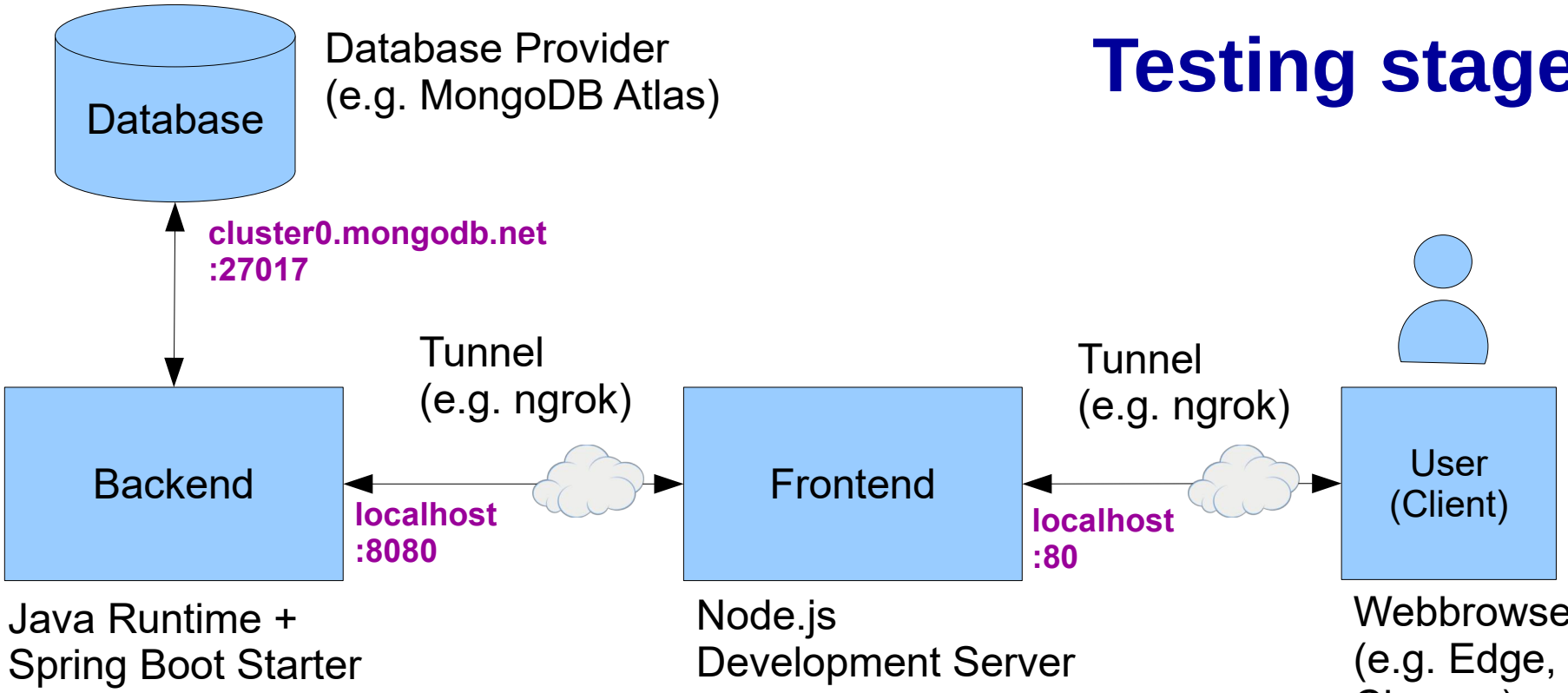
src/main/resources/.env:

```
MONGO_TYPE="mongodb"
MONGO_CLUSTER="localhost"
MONGO_DATABASE="fullstack_db"
MONGO_USERNAME="madmin"
MONGO_PASSWORD="***"
```

src/api/axiosConfig.js:

```
export default axios.create({
    baseURL: "http://localhost:8080"
});
```

Testing stage



src/main/resources/.env:

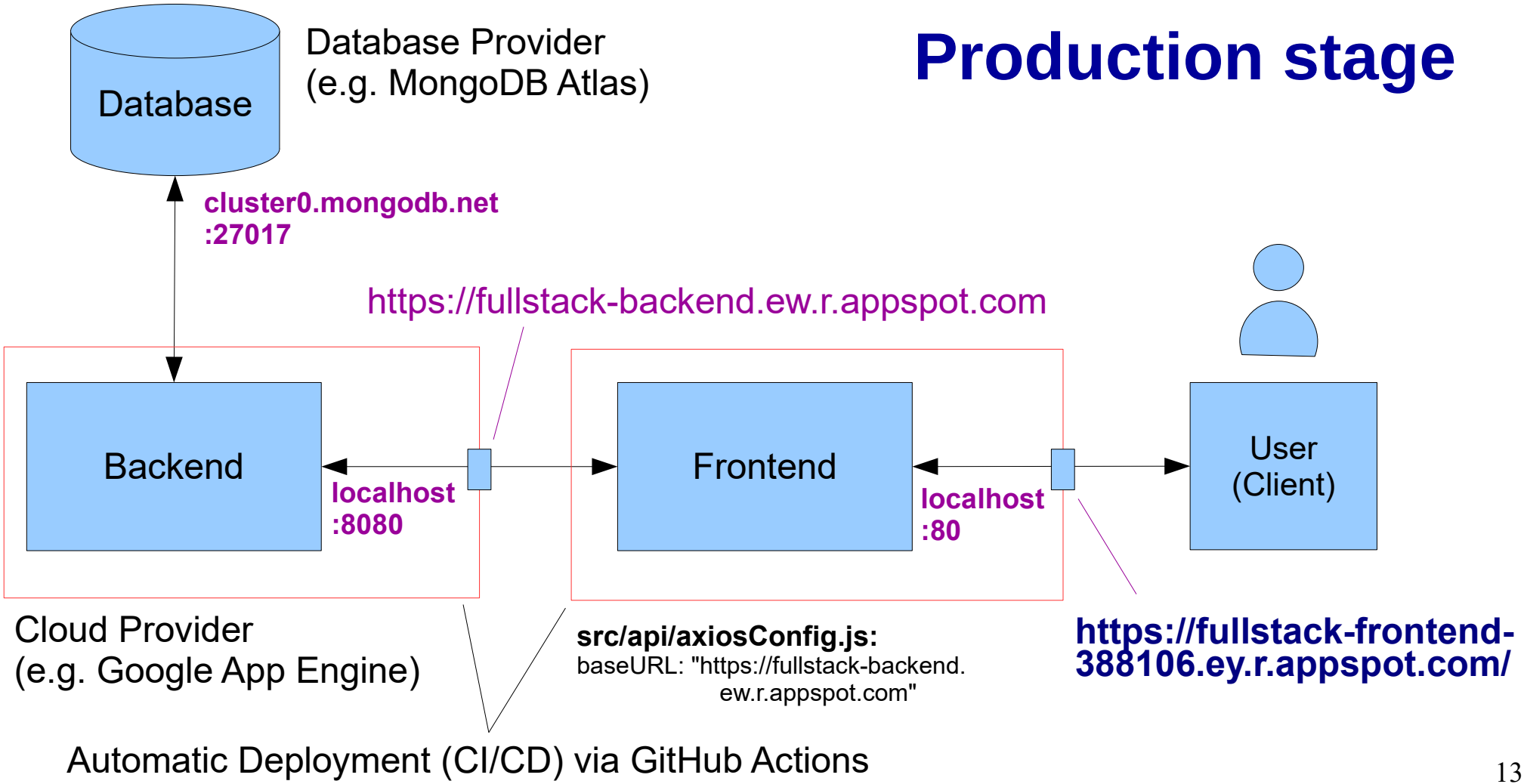
```
MONGO_TYPE="mongodb+srv"
MONGO_CLUSTER="cluster0.mongodb.net"
MONGO_DATABASE="fullstack-db"
MONGO_USERNAME="***"
MONGO_PASSWORD="***"
```

src/api/axiosConfig.js:

```
export default axios.create({
    baseURL: "https://b73b.ngrok-free.app",
    headers: {"ngrok-skip-browser-warning": "true"}
});
```

https://2a02.ngrok-free.app

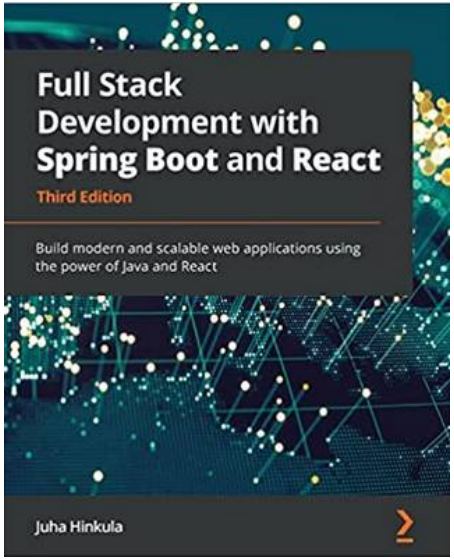
Production stage

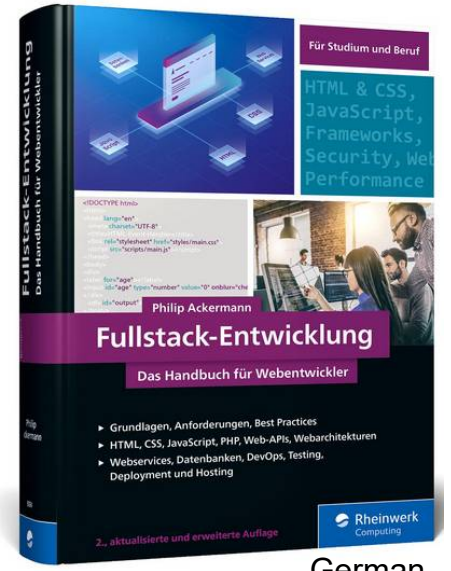


Summary

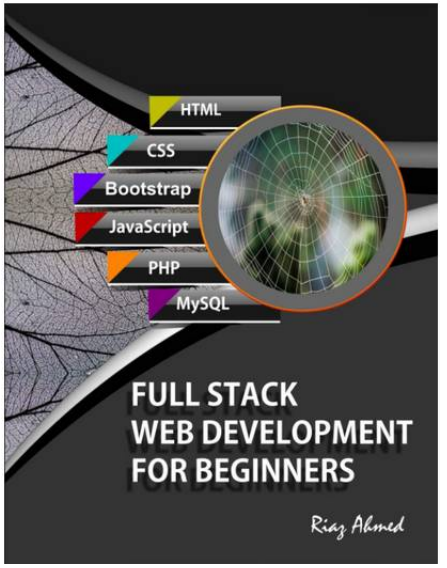
- Full stack applications consist of more than one layer: backend and frontend
- The development process has several steps
  - Defining the architecture model
  - Setting up the database (e.g. MongoDB)
  - Implementing the backend (e.g. Java, Spring Boot)
  - Implementing the frontend (e.g. Node.js, Bootstrap, React)
  - Testing and Deployment (e.g. Postman, GitHub Actions)
- Sometimes this process needs to be repeated (Agile Software Development)

Literature and Sources





German



Backend
fhsinchy/
movieist

Frontend
GavinLonDigital/
movie-gold-v1

Quiz

Join at

slido.com

#4991 646