Writing a
full stack application

Dr. Peter Dillinger
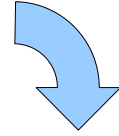peter.dillinger@email.de

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

Database

_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

Reviews

Write a Review?

Submit
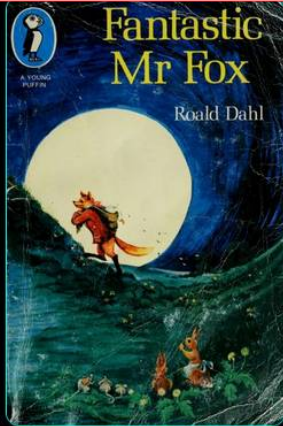
A really good book

Roald Dahl and Quentin Blake

Fantastic Mr. Fox
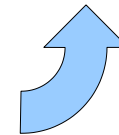
Reviews

Overview

Architecture

Database

Separation of
Concerns

Backend ⟷ Frontend ⟷ User (Client)

Full stack (more than one layer)
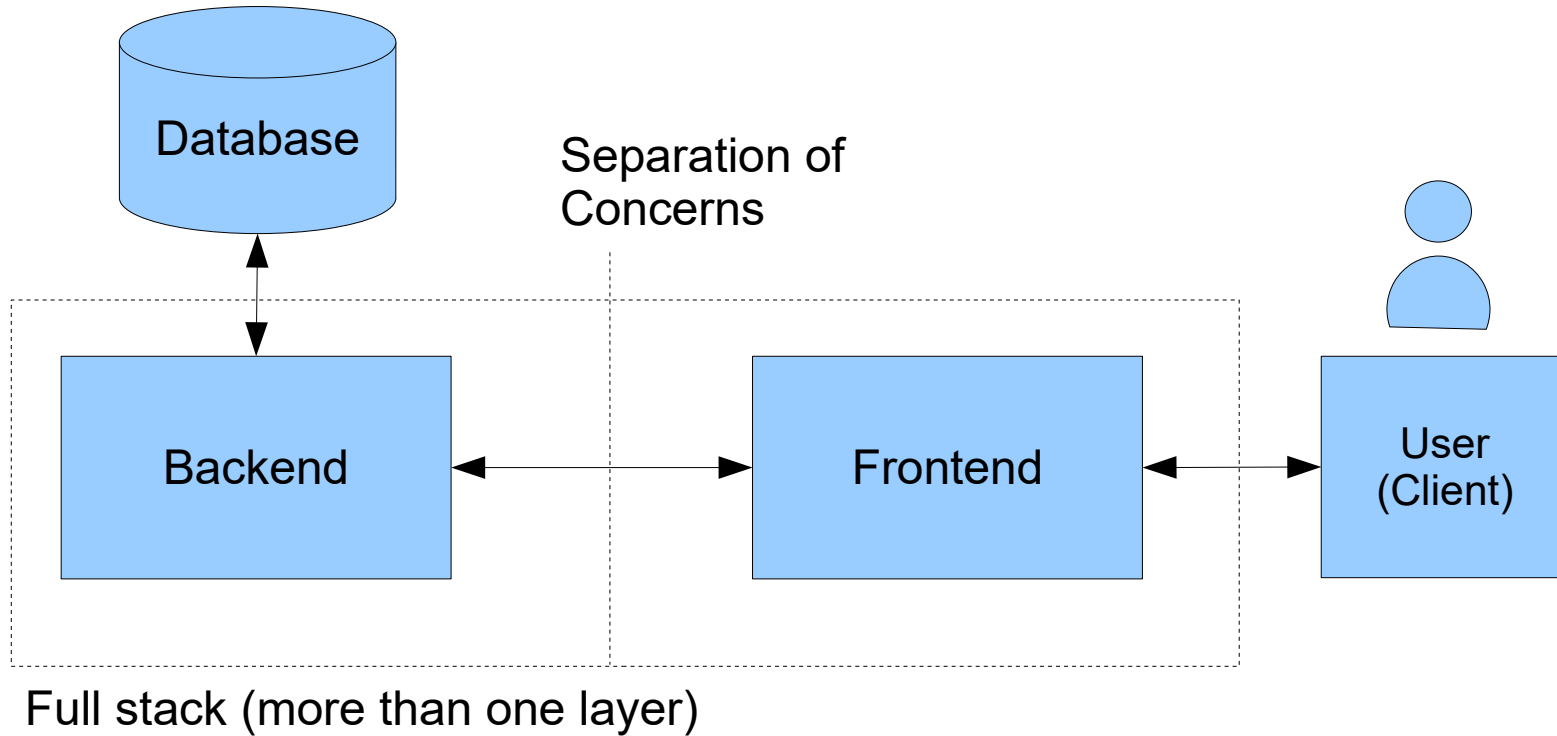
Data Management

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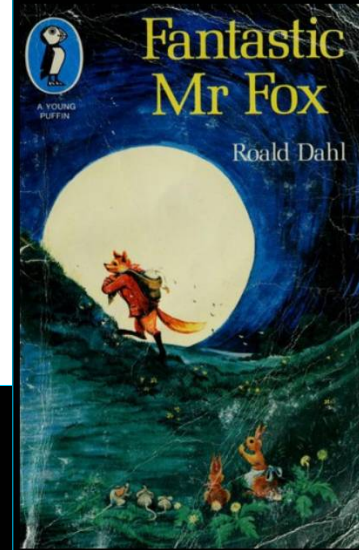
```java
@Document(collection = "books")
@Data
@AllArgsConstructor
@NoArgsConstructor
public class Book {
    @Id
    private ObjectId id;

    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

Database

MongoDB
(e.g. Virtual Box,
MongoDB Atlas)

27017

Spring Boot
Starter Data
MongoDB

Backend

8080

Spring Boot
Starter Web

Frontend

User
(Client)
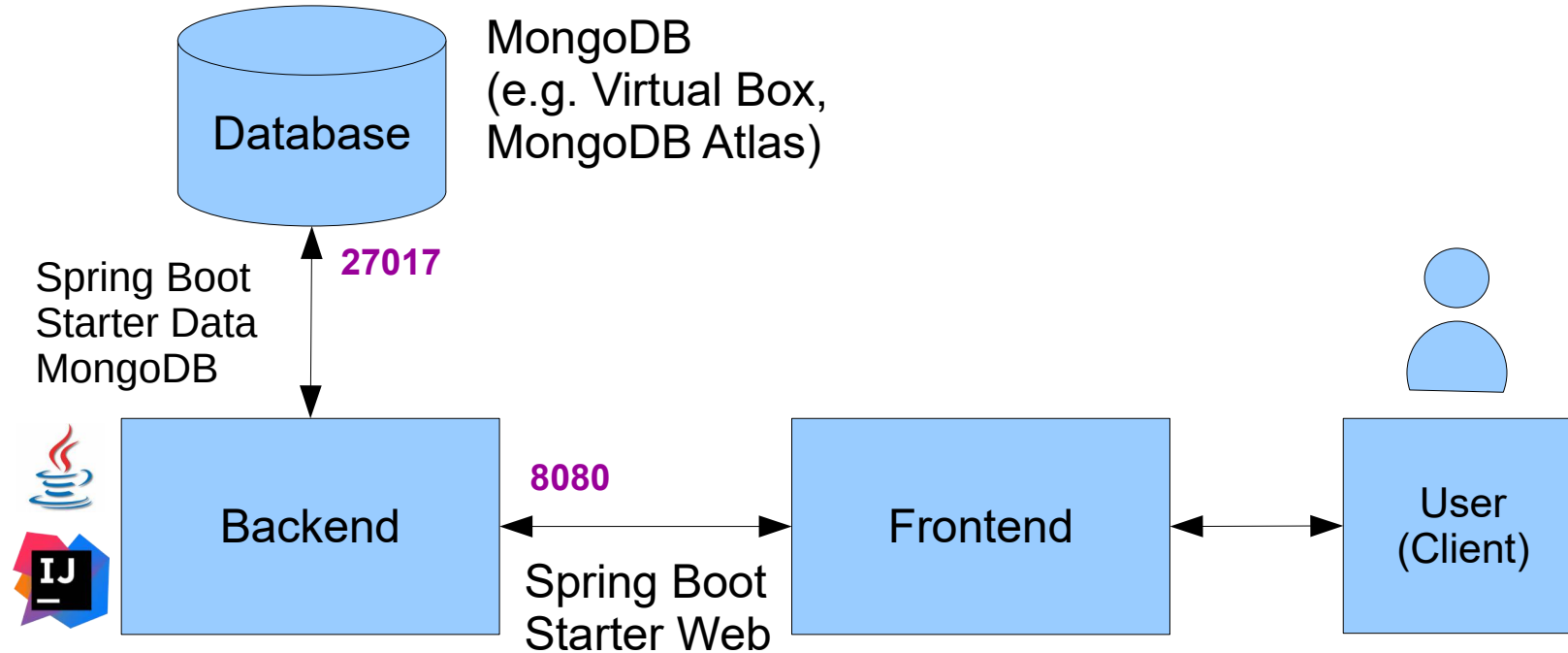
```
GET   /api/v1/books
GET   /api/v1/books/{isbn10}
GET   /api/v1/reviews
POST /api/v1/reviews, {isbn10,reviewText}
```

6

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

7

Frontend

Database

Backend

Frontend

Bootstrap
Menubar

80

User
(Client)

Axios
(HTTP client)

React
Carousel

Frontend (with light theme)

Bootstrap Menubar

HTTP Navigation

div: book-card-container

React Carousel
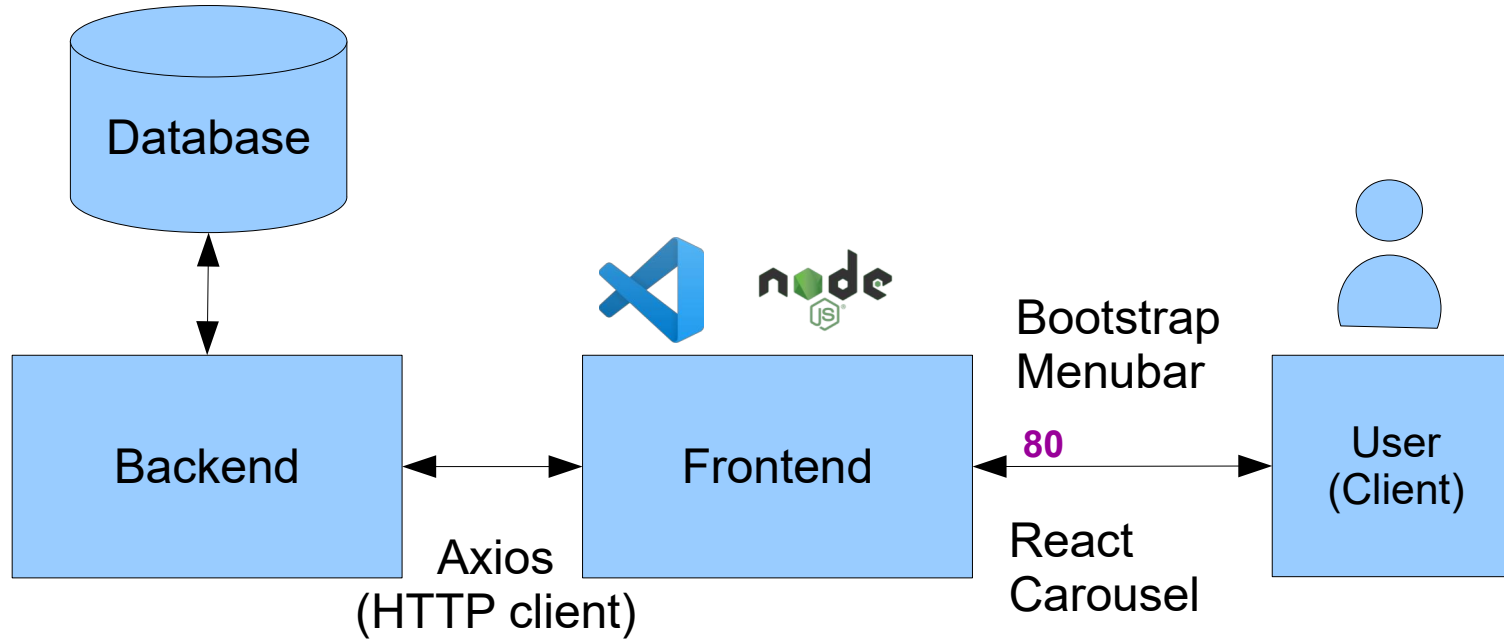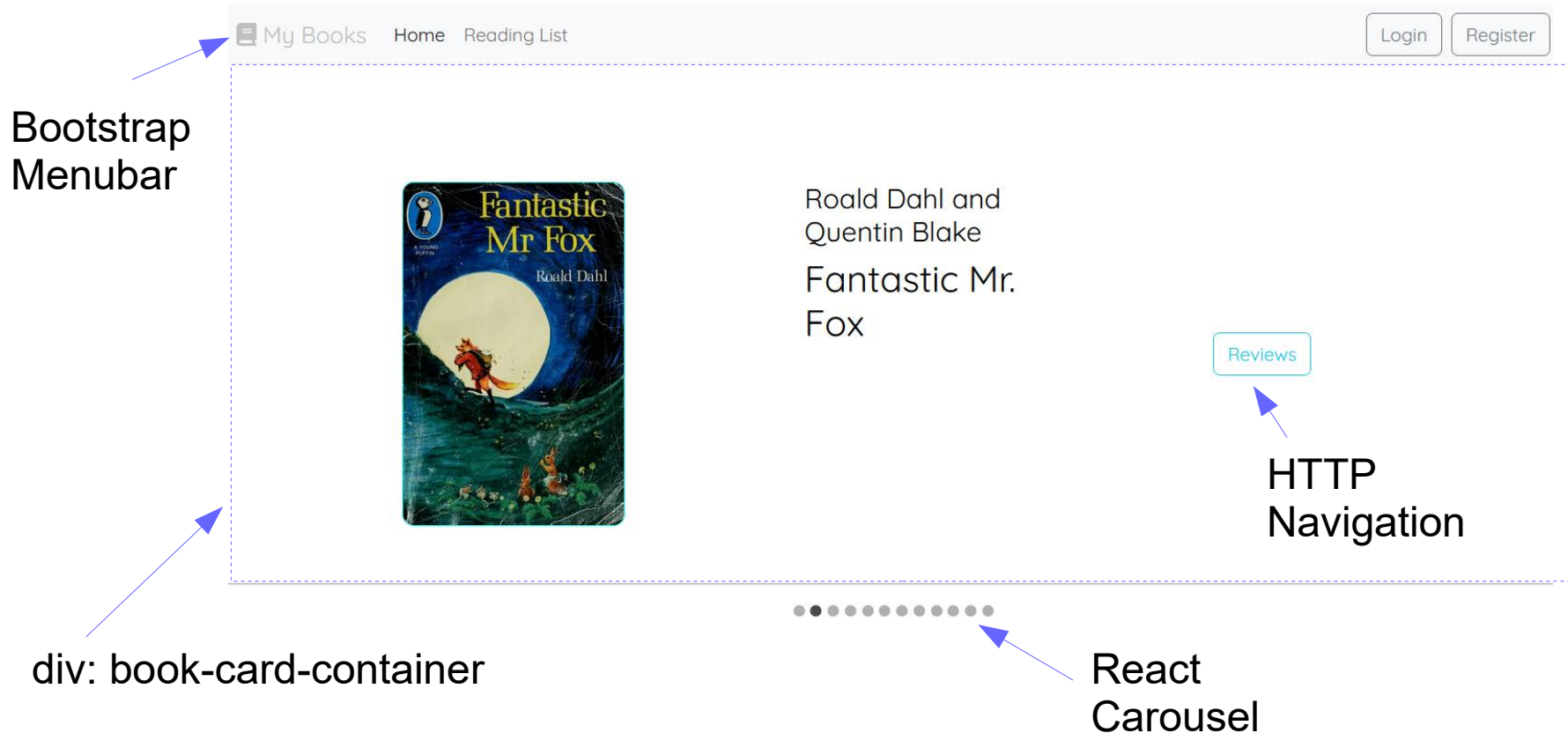
Frontend (with light theme)

Reviews
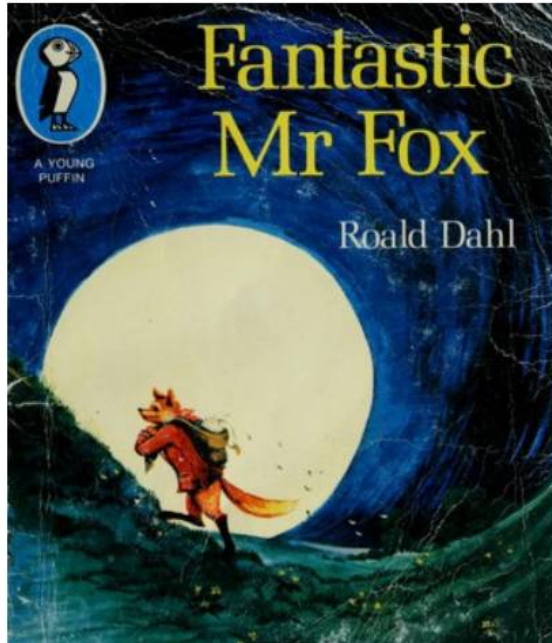
React ReviewForm

Write a Review?

Submit

Really good book!

Is it working now?

Nice :)

Row/Col Table

Virtual Box +
Docker +
MongoDB container

Development stage

Database

**VM
:27017**

NAT

**localhost
:27017**

Backend

**localhost
:8080**

Frontend

**localhost
:80**

User
(Client)

Java Runtime +
Spring Boot Starter

Node.js
Development Server

Webbrowser
(e.g. Edge, Chrome)

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"
});

http://localhost

Testing stage

Database Provider
(e.g. MongoDB Atlas)

Database

**cluster0.mongodb.net
:27017**

Tunnel
(e.g. ngrok)

Tunnel
(e.g. ngrok)

Backend

**localhost
:8080**

Frontend

**localhost
:80**

User
(Client)

Java Runtime +
Spring Boot Starter

Node.js
Development Server

Webbrowser
(e.g. Edge,
Chrome)

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**

12

Production stage

Database Provider
(e.g. MongoDB Atlas)

cluster0.mongodb.net :27017

https://fullstack-backend.ew.r.appspot.com

Database

Backend

localhost :8080

Frontend

localhost :80

User
(Client)

Cloud Provider
(e.g. Google App Engine)

src/api/axiosConfig.js:
baseURL: "https://fullstack-backend.
ew.r.appspot.com"

**https://fullstack-frontend-
388106.ey.r.appspot.com/**

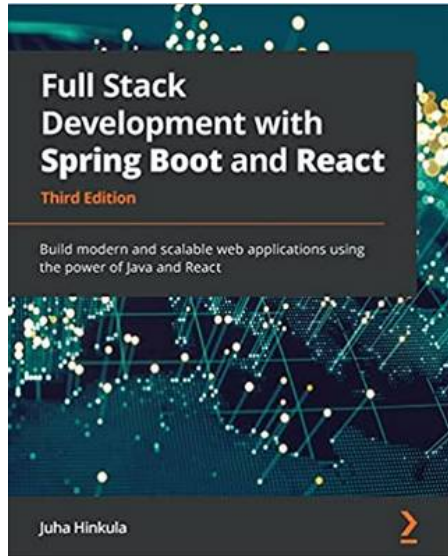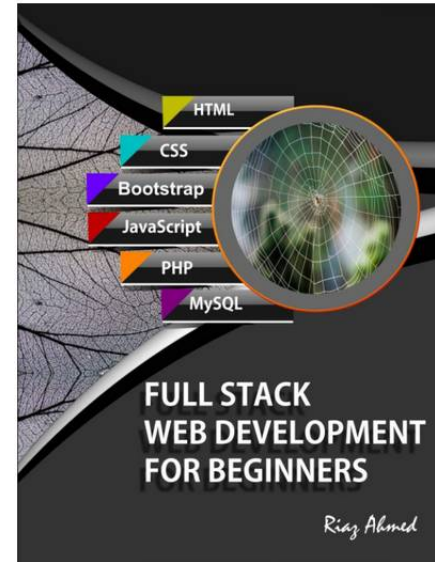Automatic Deployment (CI/CD) via GitHub Actions

13

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)

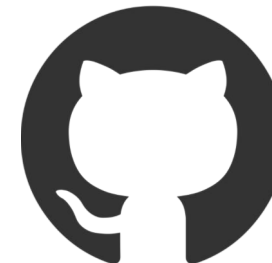- Sometimes this process needs to be repeated (Agile Software Development)

Literature and Sources

Full Stack Development with Spring Boot and React
Third Edition
Build modern and scalable web applications using the power of Java and React
Juha Hinkula

Für Studium und Beruf
HTML & CSS, JavaScript, Frameworks, Security, Web Performance
Philip Ackermann
Fullstack-Entwicklung
Das Handbuch für Webentwickler
► Grundlagen, Anforderungen, Best Practices
► HTML, CSS, JavaScript, PHP, Web-APIs, Webarchitekturen
► Webservices, Datenbanken, DevOps, Testing, Deployment und Hosting
2., aktualisierte und erweiterte Auflage
Rheinwerk Computing

German

HTML
CSS
Bootstrap
JavaScript
PHP
MySQL
FULL STACK WEB DEVELOPMENT FOR BEGINNERS
Riaz Ahmed

FULL STACK WEB DEVELOPMENT
Everything Beginners to Expert Guide on Modern Full-Stack Web Development Using Modern Web Development Tools
fullstack FULLFILLMENT
SAMMIE SMITH

Full Stack Development
Java
2:40:48

Backend
fhsinchy/
movieist

Frontend
GavinLonDigital/
movie-gold-v1

15

Quiz

Join at

slido.com

#4991 646