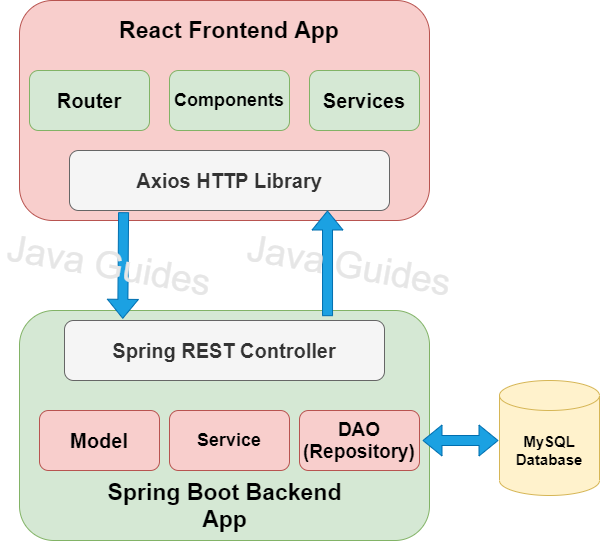
1. Spring MVC REST + React Integration **Employee Management Application**

develop a simple full-stack web application that is a basic **Employee Management Application**using React, Spring MVC REST.

The back-end server uses Spring MVC REST with Spring Web MVC for REST APIs and Spring Data JPA for interacting with the MySQL database. The Front-end side is made with React, React Router, Axios & Bootstrap.

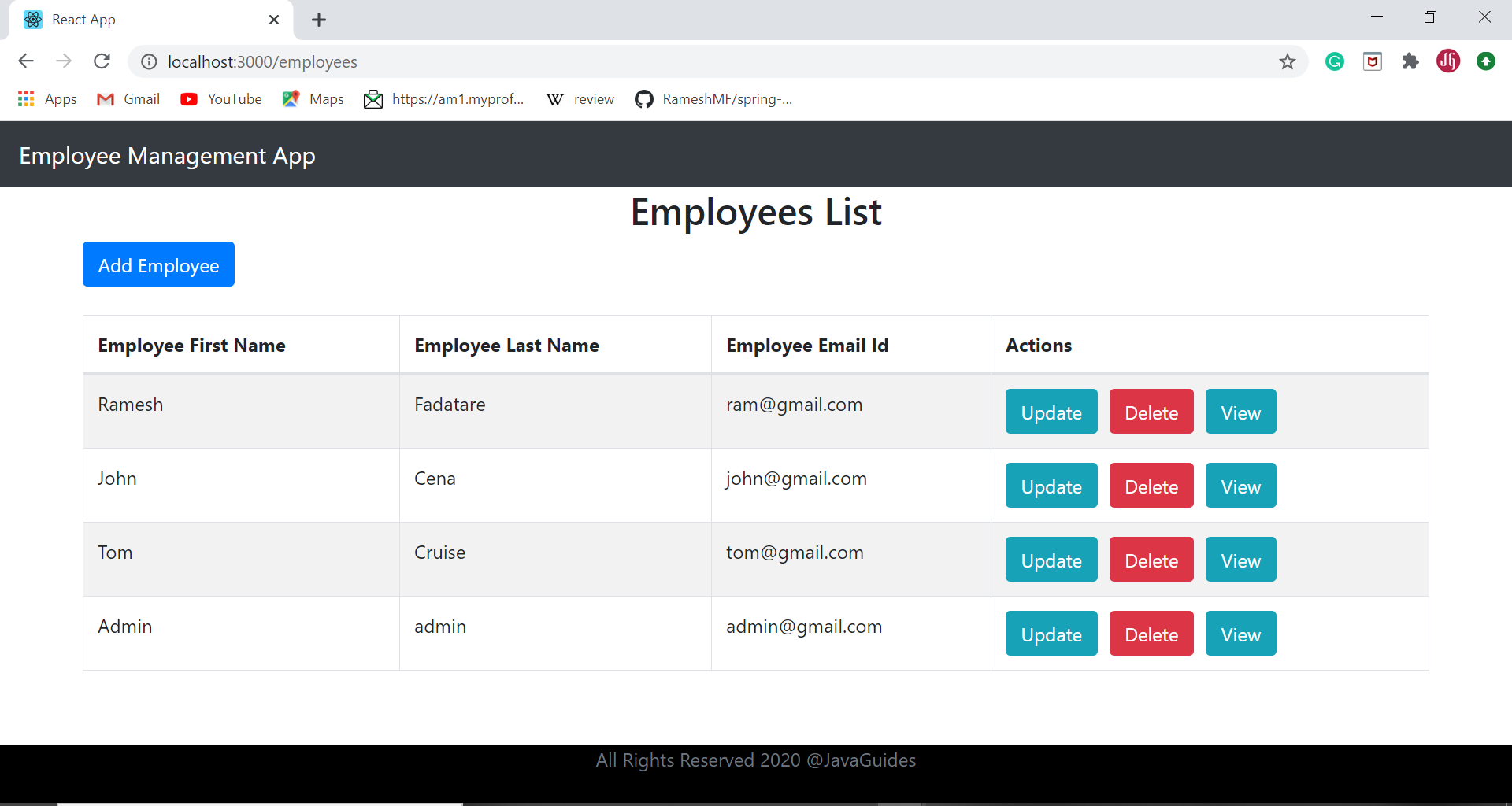
**Spring core Spring MVC REST React Full-Stack Architecture**

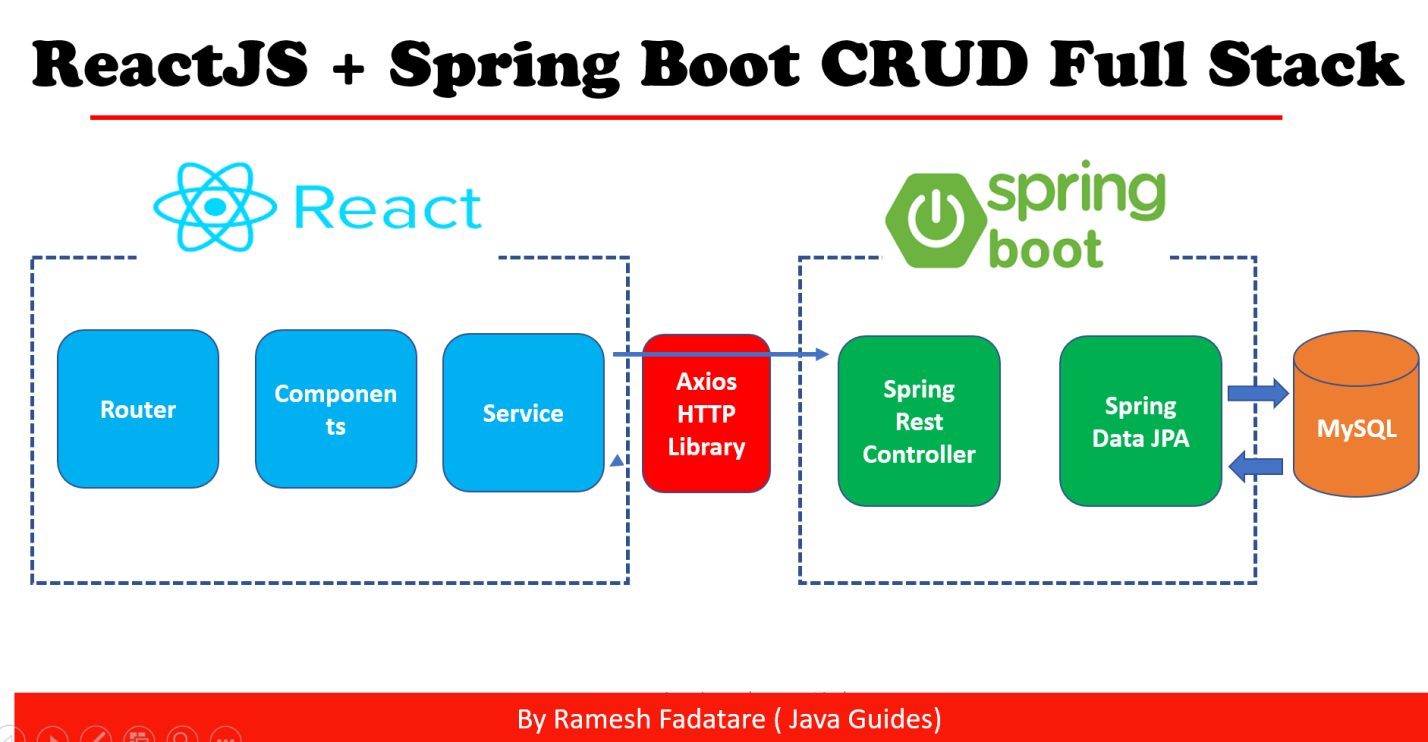


# What we will build?

We will build a full-stack web application that is a basic **Employee Management Application with**CRUD features:

* Create Employee
* List Employee
* Update Employee
* Delete Employee
* View Employee





# Tools and Technologies Used

## Server-side tools and technologies used

* Spring MVC REST
* SpringData JPA ( Hibernate)
* Maven 3.2 +
* JDK 1.8
* Embedded Tomcat 9.0
* MySQL Database
* POSTMAN

## Client-side tools and technologies used

* React
* Modern JavaScript (ES6)
* NodeJS and NPM
* VS Code IDE
* Create React App CLI
* Bootstrap 4.5 and Axios HTTP Library

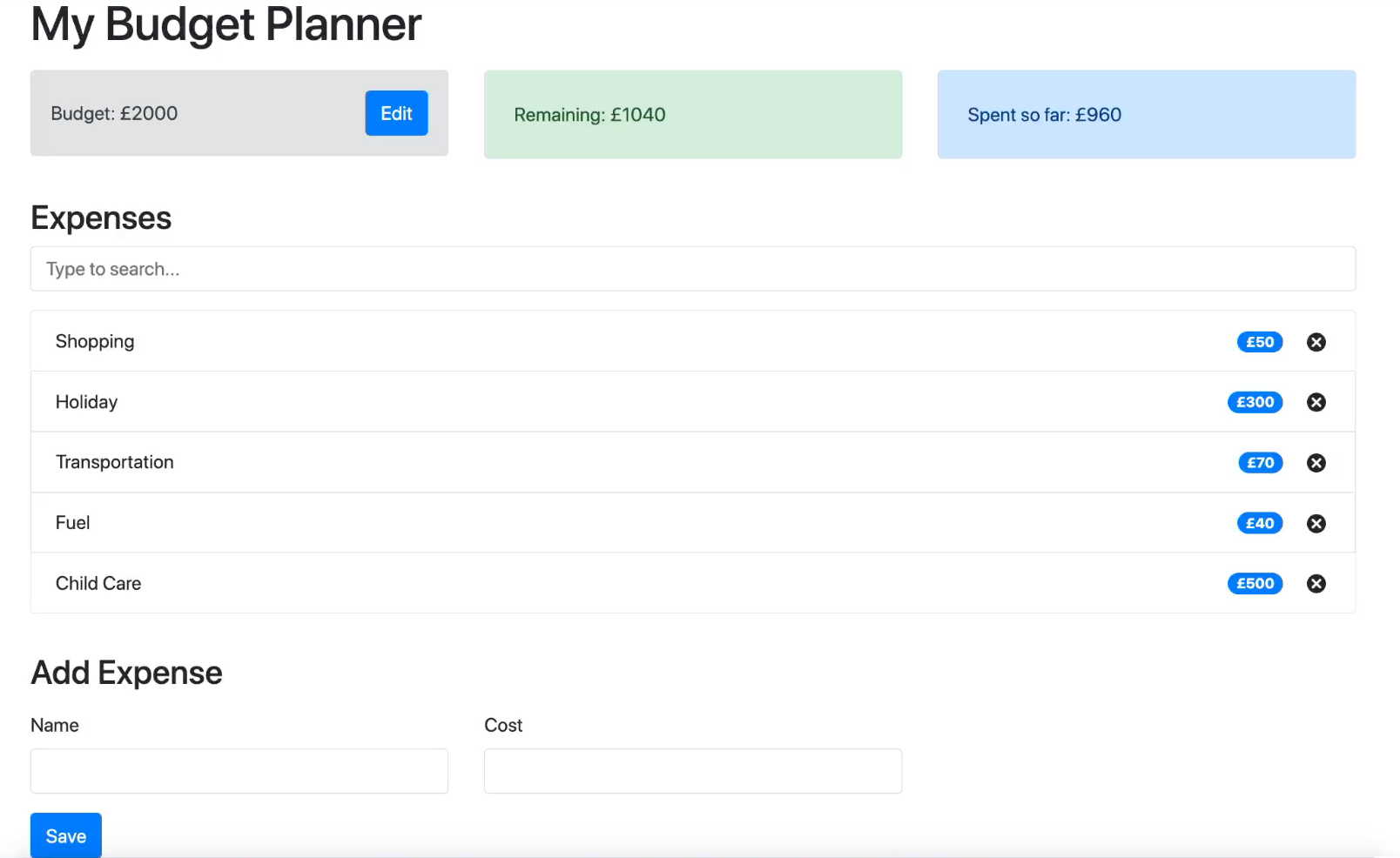
React project in Hooks

## 2. Build an Expense Tracker / Budget Tracker using React Hooks and Context API and Redux

to create an Expense Tracker / Budget Tracker Application using React Hooks and Context API and Redux



* how break down a UI into React components
* how to work with state using the Context API
* actions, reducers, and the dispatch function
* how to use hooks

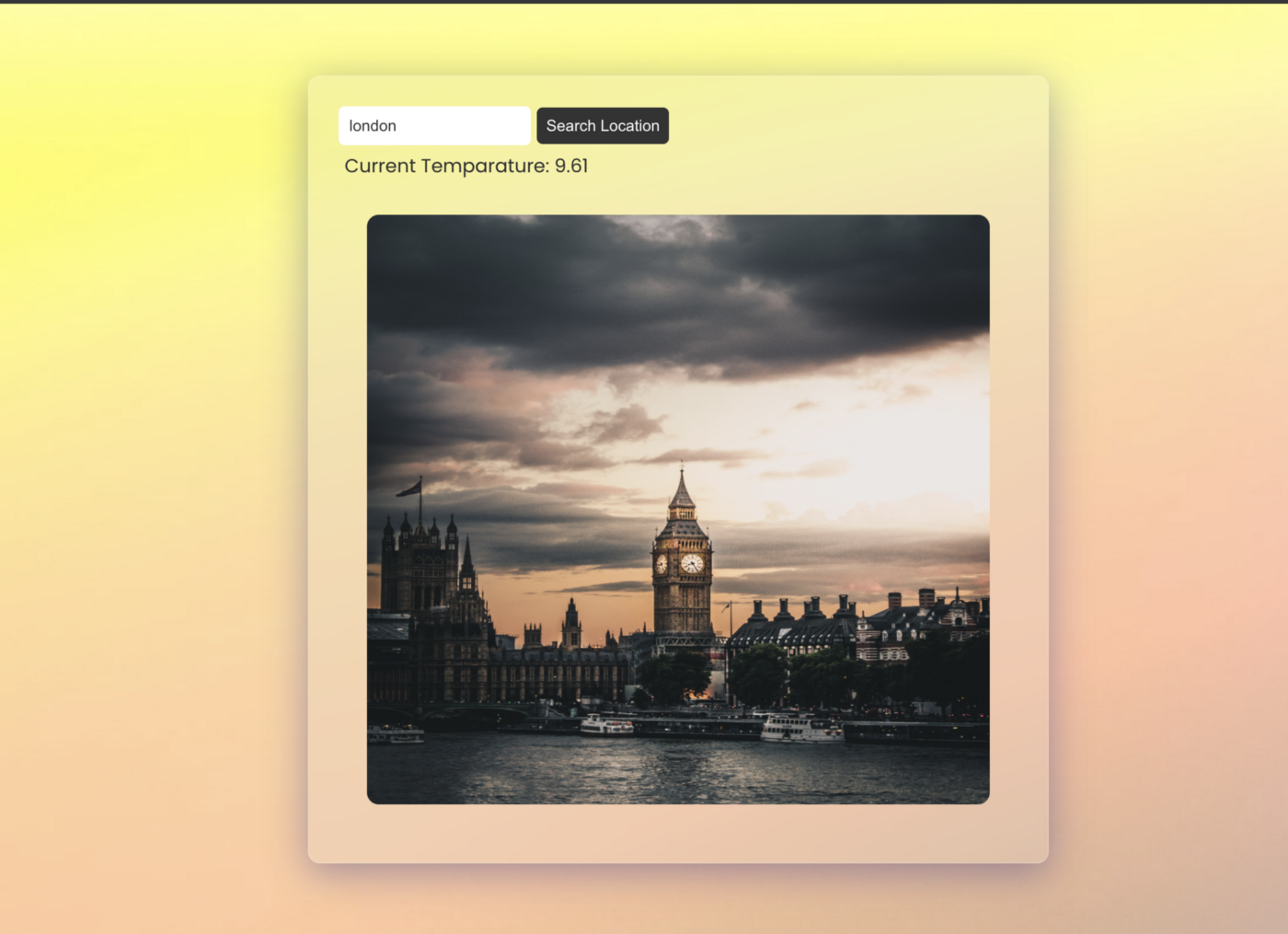


The user can:

* Add expenses which have a name and a cost
* Remove expenses
* View how much of their budget is remaining
* View how much they've spent so far
* (Challenge) Edit budget
* (Challenge) Search expenses

# 3. Weather App With React JS

Using Hooks



4. Library Management System in Java Spring MVC MYSQL 8

Objective/ Vision

A library management software where admin can add/view/delete librarian and librarian can add/view books, issue, view issued books and return books.

Users of the System

1. Admin
2. Libraian

Functional Requirements

**1. Admin**

1. Can add/view/delete librarian
2. Can logout

**2. Librarian**

1. Can add/view books
2. Can issue books
3. View issued books
4. Return Books
5. Can logout

Phases and Technologies

Phase 1. MANDATORY

Console based project (using main method) using JDBC JAVA OBJECT ORIENTED ONCEPTS

Exception Handling & COLLECTIONS MY SQL 8.0

BUSINESS LOGIC : JAVA COLLECTIONS EXCEPTION HANDLING

Front end : NIL Console based application

BACK END : MYSQL

Phase 2 MANDATORY

Web application project using Spring core Spring MVC and Spring JDBC or Hibernate

MYSQL

Front end JSP

back end spring MVC mysql spring JDBC HIBERNATE

