

A Project Report
On
“Workshop Management System”

Submitted By

Mr. Nishant Jadhav

Submitted To



NaranLala College of Professional and Applied Sciences,
Veer Narmad South Gujarat University, Surat.

Year: 2022-2023



NARANLALA
COLLEGE OF PROFESSIONAL & APPLIED SCIENCES
BHAGVATI SANKUL, NEAR ERU CHAR RASTA,
NAVSARI – 396 450

CERTIFICATE

This is to certify that **Mr. Nishant J. Jadhav**, Exam No. _____ student of **B.C.A. 5th Semester** of our college have successfully prepared and submitted Project Report on “**Workshop Management System**” as a partial fulfillment for the course of the **Bachelor of Computer Application (Minor Project Sem. V)** during the academic year **2022-2023**.

DATE

Guide: Dr. A.B. PATEL

Dr. S. M. NAIK
(PRINCIPAL, NLCPAS)

Dr. A. B. PATEL
(DEPT. HEAD, BCA)

(EXTERNAL EXAMINER)



Workshoply

Acknowledgment

Dear Reader,

I sincerely feel that I am not the only one to be credited for this Web App. This mini project work is an integrated effort of all self-study from Online Forums, YouTube tutorials, Official Docs, StackOverflow, and all those concerned, by whose cooperation and practical guidance; I achieved its completion.

It is a curriculum to undergo project work in any system. I preferred the “**WORKSHOP MANAGEMENT SYSTEM**” at Naranlala College of Professional and Applied Sciences, Navsari.

I would like to take this opportunity to express my gratitude to all the faculty members of the Computer Science department, Principal **Dr. Sunil Naik**, Head of Department **Dr. Ashish Patel**, and special thanks to **Mr. Chirag Mewada** and **Ms. Neha Parmar**.

Yours truly,

NISHANT JADHAV

(github.com/nisoojadhav)

Abstract

The report reflects the journey of making a Full Stack MERN Application. It incorporates my learnings from errors, opening it for open source, hosting it on the web, and working with dynamic web technologies and libraries. I have gained knowledge of not only programming but about hosting on the web, finding vulnerabilities, SEO, designing paradigms, typography, and working with DB on the cloud.

I would like to thank my 3 FPGs(Friend, Philosopher, Guide):

1) **Mr. Chirag Mewada** (who stated me to learn Javascript and React/Angular right in Sem 4).

(<https://www.linkedin.com/in/chirag-mewada-csm%C2%AE-90015213/>)

2) **Sumit Patil** (who expressed me learning Javascript and Mongo DB when I was in Sem 3(I didn't take it seriously though I needed to learn it at the end of Sem 4)).

(<https://www.linkedin.com/in/iamsumitpatil/>)

3) **Zaid Shaikh** (Senior Web Developer at Prism IT Systems, who helped me learn React JS, and MERN and keeps me updated about what tech stack I learn next).

(<https://www.linkedin.com/in/zaid-shaikh-391aa1138/>)



Contents

| | | |
|----------|---|----|
| | Abbreviations | 1 |
| 1 | Introduction | 2 |
| | 1.1 Project Details | 3 |
| | 1.2 Purpose | 4 |
| | 1.3 Project Scope | 4 |
| | 1.4 Limitations | 5 |
| 2 | System Requirements | 6 |
| | 2.1 Hardware Requirements | 6 |
| | 2.2 Software Requirements | 6 |
| | Building Local Version | 7 |
| 3 | System Analysis | 9 |
| | 3.1 Requirement Analysis | 9 |
| | 3.2 Feasibility Studies | 13 |
| | 3.3 Main Module of System | 14 |
| 4 | User Privileges | 15 |
| | 4.1 Admin: (CRUD) Workshops, Attendees | 15 |
| | 4.2 Other Users: | 22 |
| | Read(Search), Contact, Register for Workshop | 22 |
| 5 | Tools and Technologies | 26 |
| | 5.1 Overview | 26 |
| | 5.2 Why React JS? | 27 |
| | 5.3 CSS and Bootstrap | 28 |

| | | |
|----------|-------------------------------------|-----------|
| 5.4 | Node Modules | 29 |
| 5.5 | Deployment to Heroku | 31 |
| 5.6 | MongoDB Atlas | 35 |
| 5.7 | Database Schema | 36 |
| 5.8 | REST API | 38 |
| 6 | System Diagrams | 39 |
| 6.1 | System Flowchart | 39 |
| 6.2 | ER Diagram | 40 |
| 6.3 | Data Flow Diagram | 42 |
| 6.4 | Use Case Diagram | 44 |
| 7 | Scope of further development | 45 |
| 8 | Important Links | 46 |
| 9 | References | 47 |

Abbreviations

| | |
|--------------|---------------------------------------|
| API | Application Programming Interface |
| BSON | Binary JavaScript Object Notation |
| CLI | Command Line Interface |
| HTTP | Hypertext Transfer Protocol |
| JSON | JavaScript Object Notation |
| JSX | JavaScript XML |
| MVC | Model View Controller |
| MERN | MongoDB, Express, React JS, Node JS |
| NPM | Node Package Manager |
| NoSQL | Not only Structured Query Language |
| RDBMS | Relational Database Management System |
| REST | Representational State Transfer |
| SPA | Single Page Application |
| URL | Uniform Resource Locator |

1 Introduction

The Covid 19 pandemic forced us to change the way we work. It forced us to stay detached yet connected digitally as a social-animal. Everything that can happen digitally was adapted more rapidly compared to the pre-pandemic era. Fields range from contactless payment to delivery of groceries and medicines. And from the digital admission process to virtual learning. E-learning was never such a good solution to us before the pandemic but the post-pandemic digital space boomed in the world, and a digital revolution in India was responsible for multi-billion dollar unicorns leaping from the e-learning industry. Now it's so common that it's believed the institution doesn't exist if it isn't available in any digital space. Many YouTube Channels, Online Learning BootCamps, and Online interactive virtual learning sites provide online learning, certifications, and Universities that provide distant, online graduation and post-graduation programs taking lectures via the internet.

“Workshoply - a one-stop solution for managing workshops” is an open source project for helping small institutions having hurdles in their way to providing intuitive learning for their students in the comfort of their homes.

Project Details

- This web application delivers users an intuitive way by which they can get the course details, register, and attend the workshop from the comfort of their homes.
- This web application will be a boon for the people who want to host new workshops and want other people to gain from their teachings.
- Teachers can request password details. Idle users can just register the colleague will get back to them.
- After successful registration, before seats get full, the learners will get an email for Zoom meetings or the physical location of the workshop.

Purpose

- Provides new skills to learn from the home.
- Access current running workshops.
- Reduce the cost of the offline and tedious process of hosting such workshops.
- Facility for admin to manage to Create, Read, Update, and Delete permissions for the workshops.
- Other users will have Read(Search) and Register permissions.

Project Scope

- Perfect for Institutions to serve 1000s of students.
- Register details in Google Spreadsheet, accessible through the Workshoply app.
- Teachers can focus more on teaching instead of worrying about data handling, security, hosting, and thousands of concurrent users.
- Async requests through REST API, MongoDB Atlas.

Limitations

- Not included Login, Sign up for students.
- Teachers can request a Password from the Admin, which is made with such logic that it changes every hour to prevent gaining access to the Admin Page by the unauthentic person even if he receives a password.
- Not included Virtual Meeting Feature(like Zoom).

2 System Requirements

1. Hardware Requirements

- If you want to deploy the app in the local environment:
- PC/ Laptop.
- Minimum 4 GB RAM.
- Processor 1.4 GHz 32/64-bit.
- Internet connection.

2. Software Requirements

- Windows 8.1 (minimum)
- Latest Node JS, NPM, and Git installed.
- Internet connection.
- Browser (required) with React Developer Tools (optional) installed.

Building Local Version

Prerequisites:

- Installed Node JS, npm/yarn, React JS, Git Bash(or another terminal), and a text editor(VSCode preferred).

Download:

- Node JS: <https://nodejs.org/en/download/>
- npm: <https://www.npmjs.com/package/npm>
- Yarn: <https://www.npmjs.com/package/yarn>
- React: <https://www.npmjs.com/package/react>
- Git Bash: <https://git-scm.com/downloads>
- VSCode: <https://code.visualstudio.com/>

2. Open Git Bash, get into your preferred directory.

e.g.

```
$ cd d:/projects
```

3. Clone [Workshoply](#) git repository.

```
$ git clone https://github.com/nisoojadhav/workshop.git
```

4. Get into Workshop folder

```
$ cd workshop
```

5. Install packages:

```
$ npm i  
$ cd client  
$ npm i
```

6. Start server(resides on root folder):

```
$ node server
```

7. Start project

```
$ cd client && npm start
```

8. You are good to go! Visit localhost:3000.

3 System Analysis

Requirement Analysis

MERN Stack:

1) MongoDB:

A free, open-source, cross-platform, document-oriented cloud database. Developed while keeping scalability and developer agility keeping in mind. Instead of storing data in rows and columns, it stores JSON documents in collections with dynamic schemas making it easier to store and combine data of any structure, without complex validation rules, and schemas, with flexible data access, and rich indexing functionality.

2) Express JS:

Web application framework that runs backend applications. Express runs as a module in Node JS, it can handle the routing of requests to the right parts of the application:

We use Express to perform two functions:

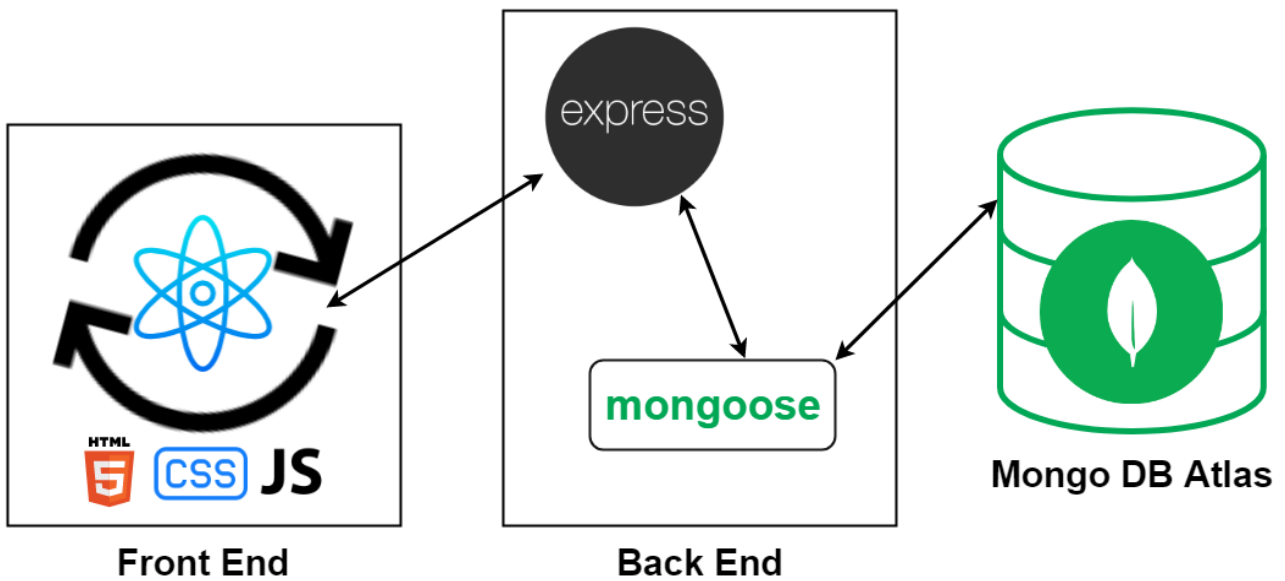
- A) Send the front-end to the remote browser when a user browses our app.
- B) Provide REST API that the front-end can access using HTTP network calls to access the database.

3) React JS: JavaScript library developed by Facebook to build interactive UI. React breaks the front end into components. Each component can hold its state that the parent can pass down to its child components, and the child can pass changes back to the parent component using callback functions.

4) Node JS: JavaScript runtime environment that runs our back-end application(via Express). Node JS is made upon Google's V8 JS engine, used in the Chrome browser. Third-party modules are installed using npm(Node Package Manager). Node JS is an asynchronous, event-driven engine where the application makes a request and then continues working on other useful tasks rather than stalling until a response. The application receives results via a callback, ensuring a lot of operations are performed in parallel when scaling applications.



Node JS Usage 2022.



3 Tier MERN Architecture

Goals for conducting Requirement Analysis:

- Identify customers' needs.
- Evaluate the system for feasibility.
- Economic and Technical analysis.
- Functions to system elements.
- Enforcing constraints.

Steps included in Requirement Analysis:

- Problem recognition
- Evaluation and synthesis.
- Specification
- Review.

Requirement Types:

1) Functional Requirements:

-Involves Input Output processes error handling.

2) Non-functional Requirements:

-Physical environment(equipment locations, multiple sites, etc.)

-Interface(electronic medium).

-User and human factors.

-Security.

-Quality assurance.

3) Validation:

- Are the requirements complete, accurate, uniform, and verifiable?

Workshoply has fulfilled all the above requirements. It is accessible via any electronic device that has a browser on it. It fulfills the requirement of helping instructors create and manage new workshops whose learners can learn from anywhere. The system runs successfully without errors. We can proceed further to the FAST technique:

FAST(Facilitated Application Specification Techniques).

The Objectives of FAST:

- Specify requirements.
- Identify the problem.
- Propose a solution.
- Negotiate between approaches.

Feasibility Studies

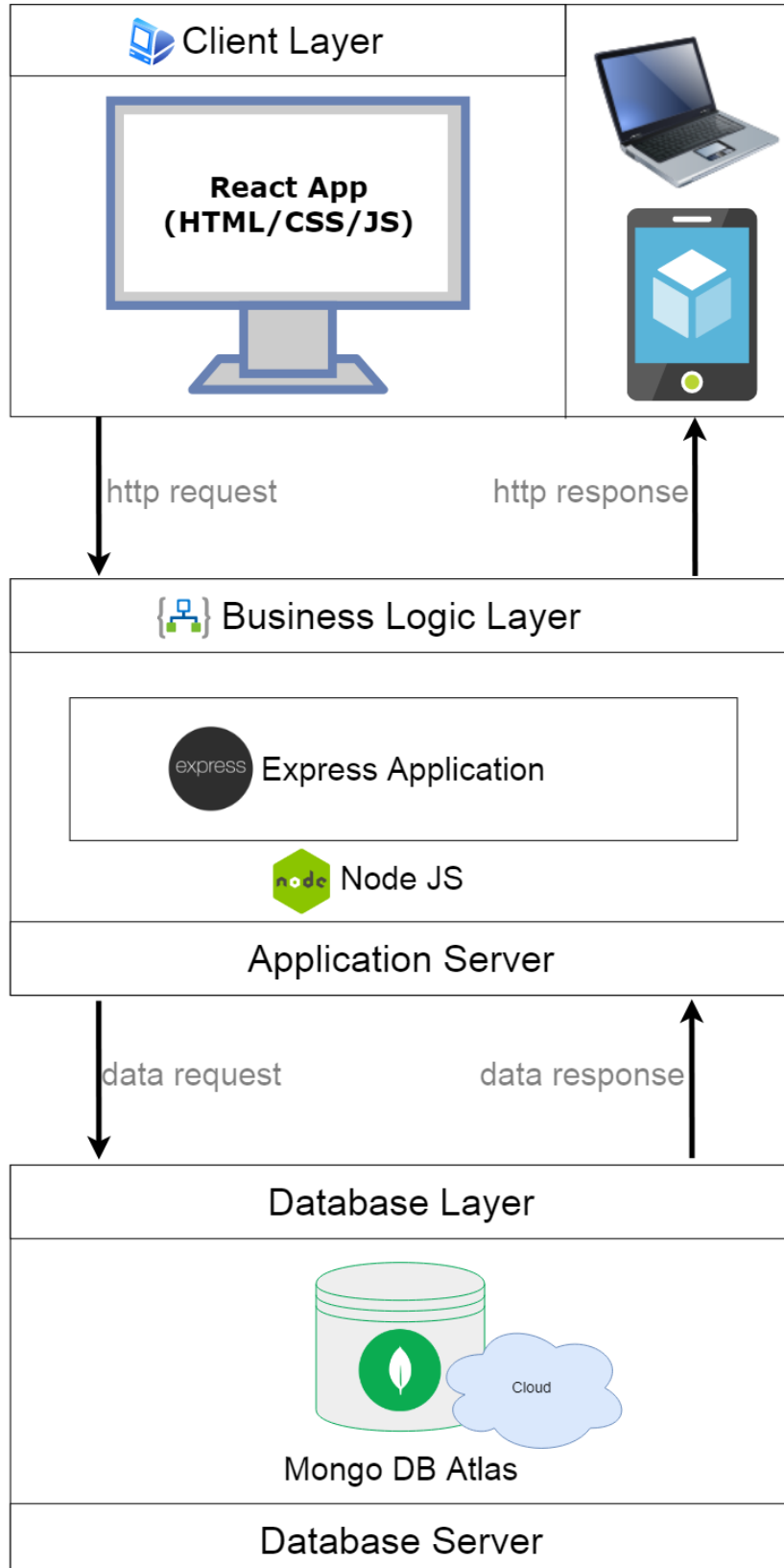
Workshoply is feasible, it does the task for which it was designed, as a learner I have incorporated all the simple tasks that are performed. As an open-source project, I foresee including many amenities inside it as time permits. **Operational and Technical feasibility** is covered as it's secured, cloud-based, and hosted on the web, so serves its purpose.

As an open-source project, **economic feasibility** is already satisfied and the product is free to use.

Schedule feasibility is also kept in mind as the testing environment is hosted on a dummy site and the actual production build is hosted on the main site.

Main Module of System

Main Module of System Overview:



4 User Privileges

4.1 Admin(CRUD Operations - Workshops and Attendees)

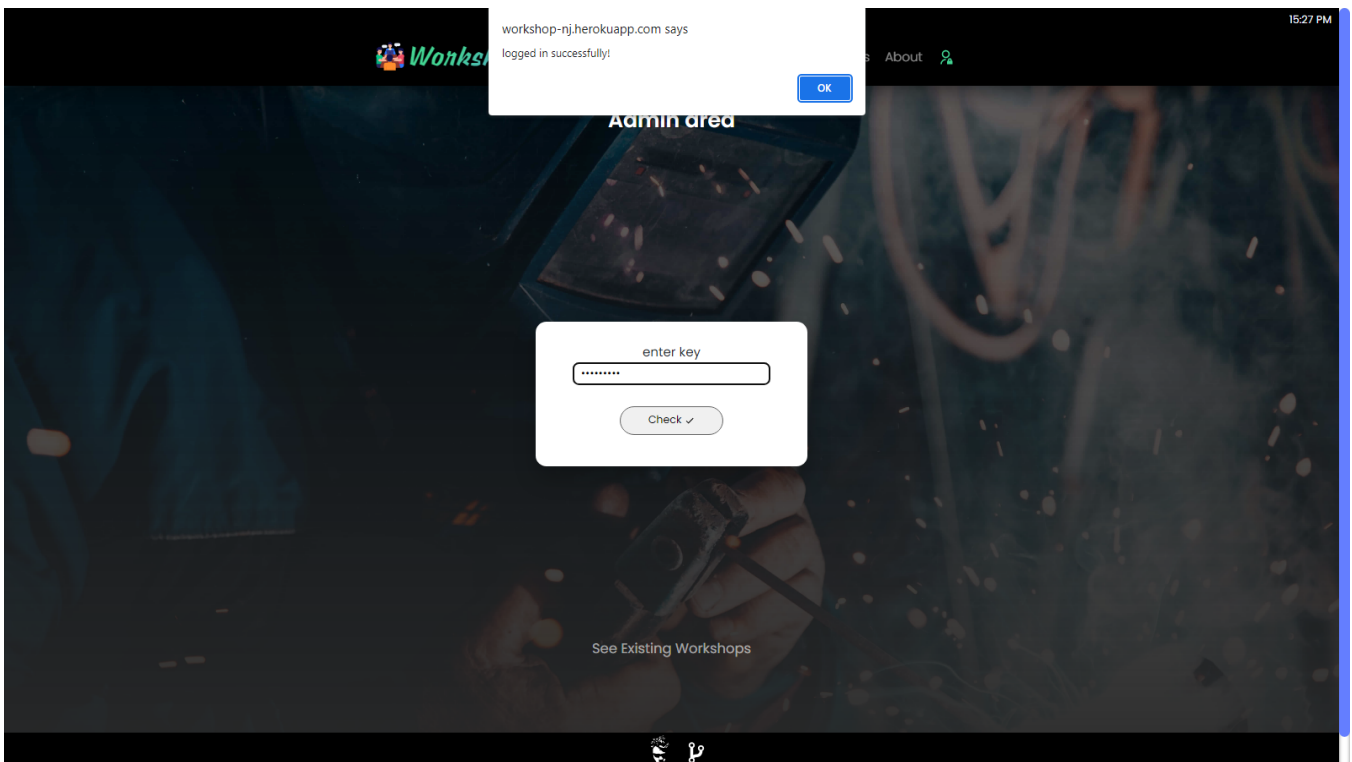
1) Workshops

Creating Workshops

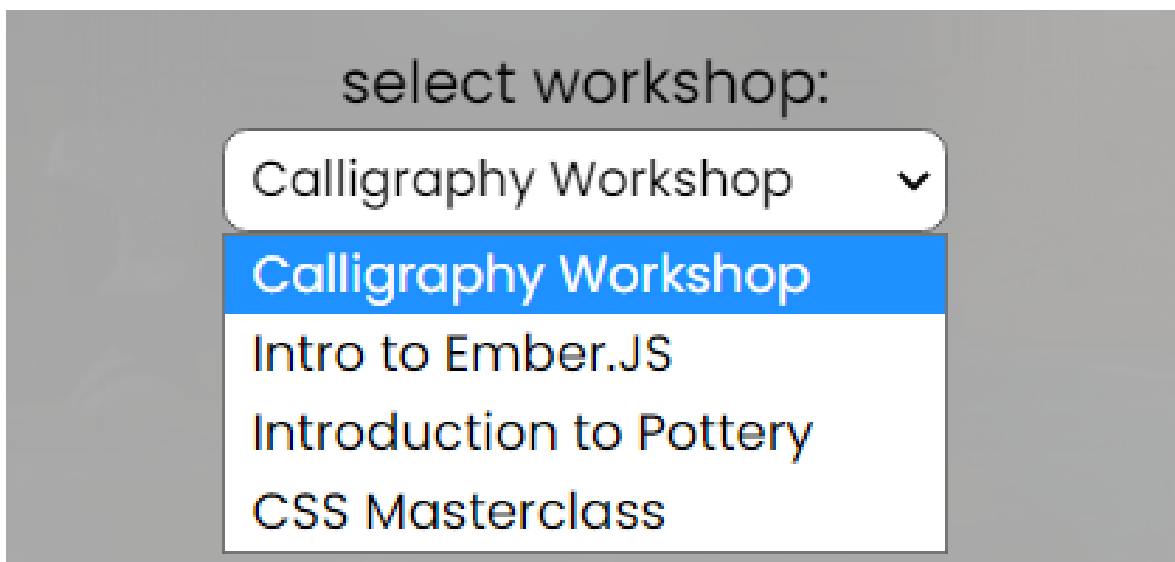
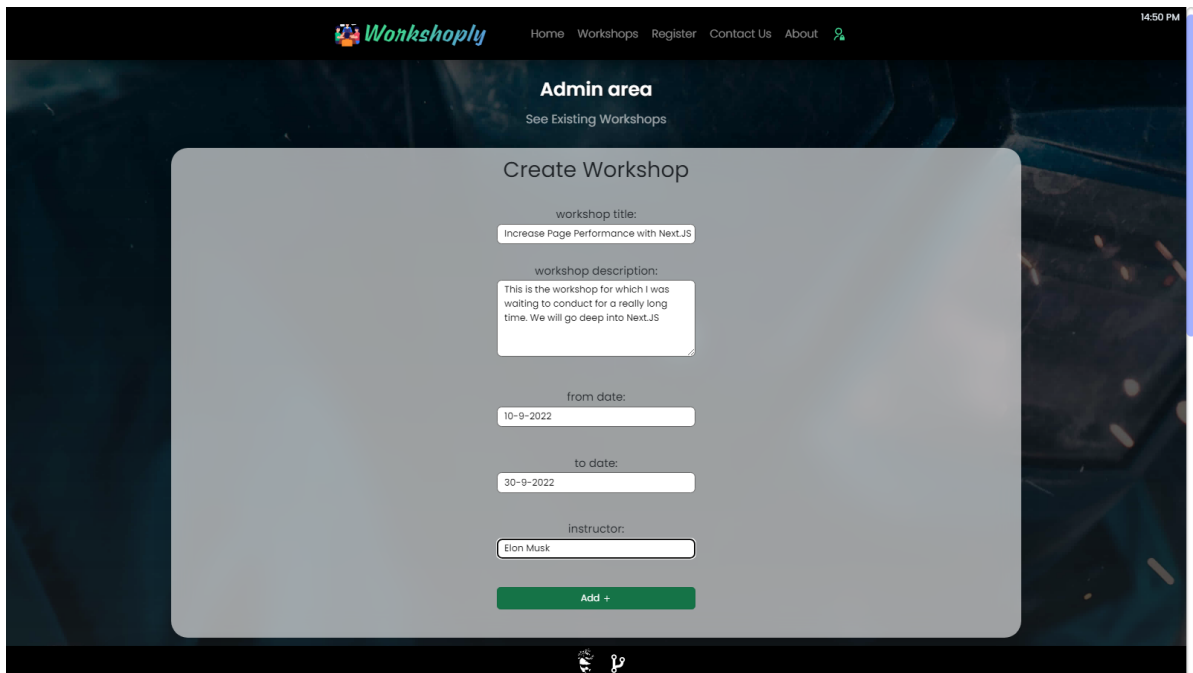
- The Admin can create workshops by visiting and successfully logging in:

<https://workshop-nj.herokuapp.com/create>

Logging In

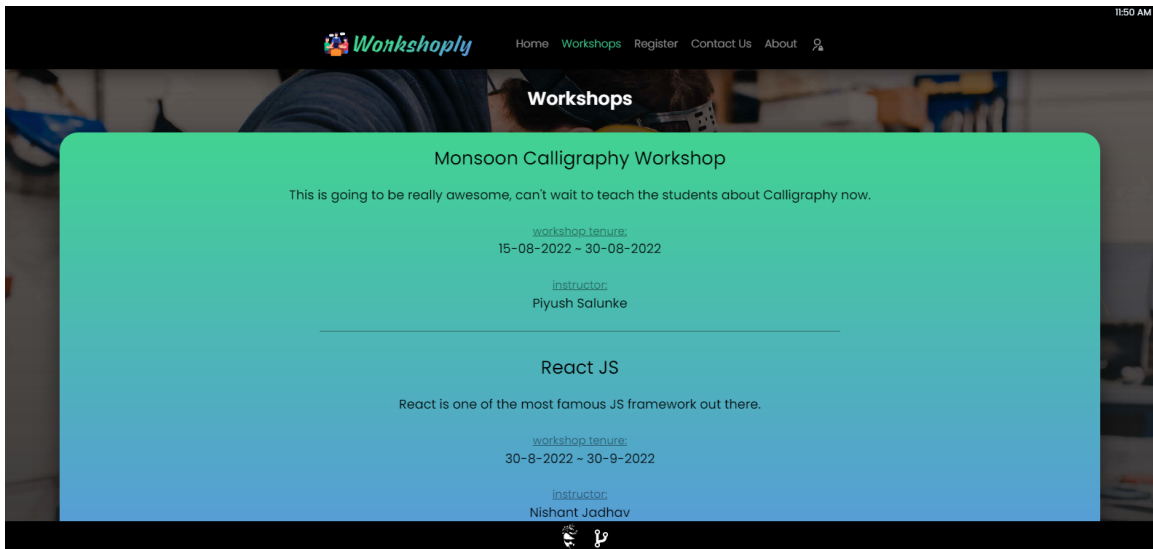


- Creating a workshop will insert the details into MongoDB Cluster and asynchronously append the new workshop at the end of the workshops page and in the dropdown on Register Form.

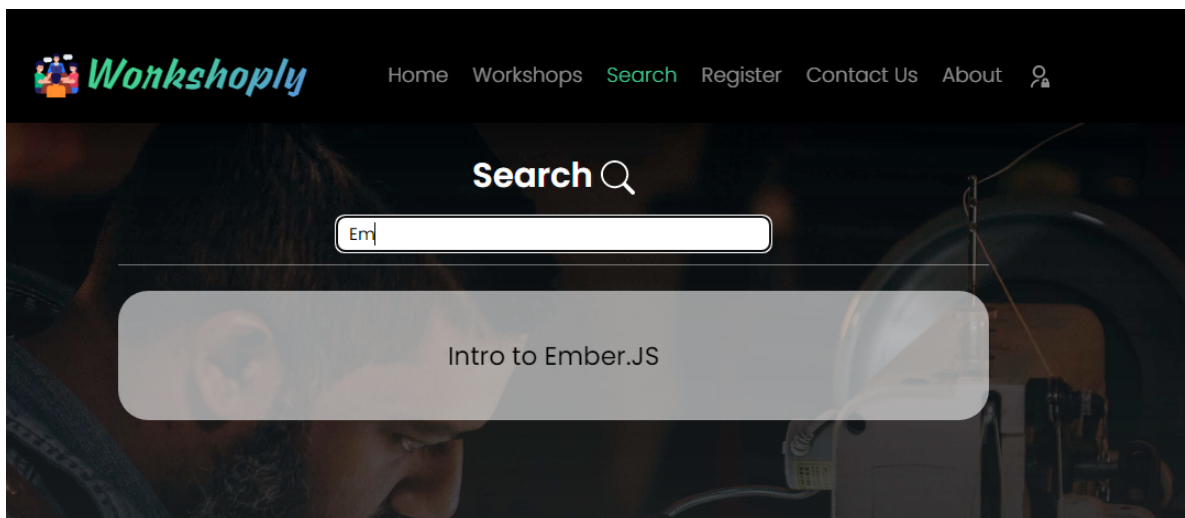


Reading Workshops

- Reading data from the MongoDB Cluster(Atlas) is carried out asynchronously, as we can see in Workshops, and the dropdown of Register.

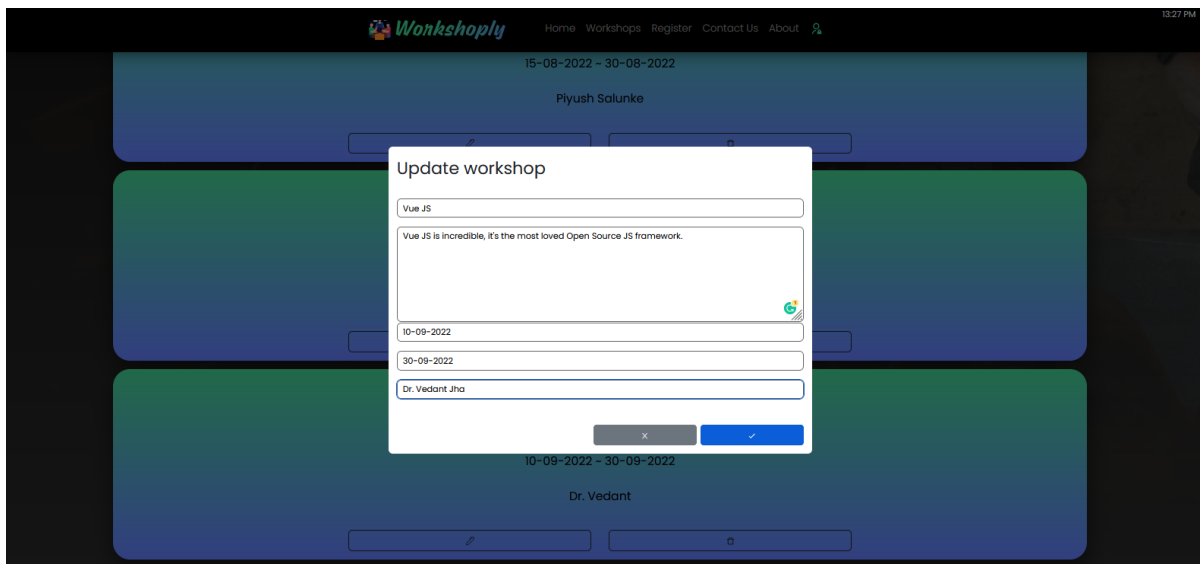


Search



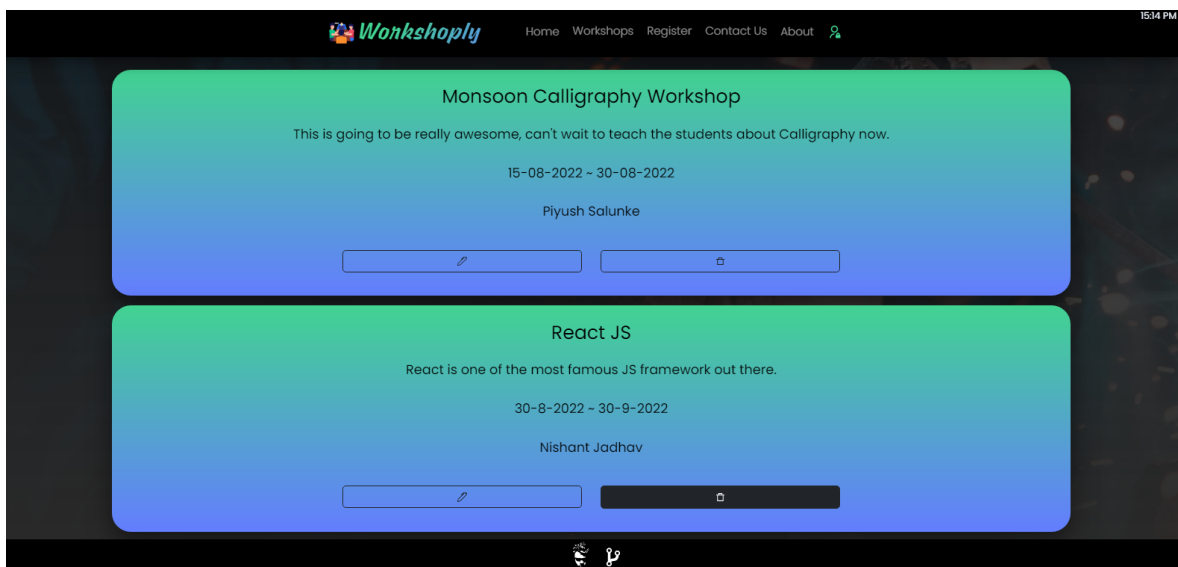
Updating Workshops

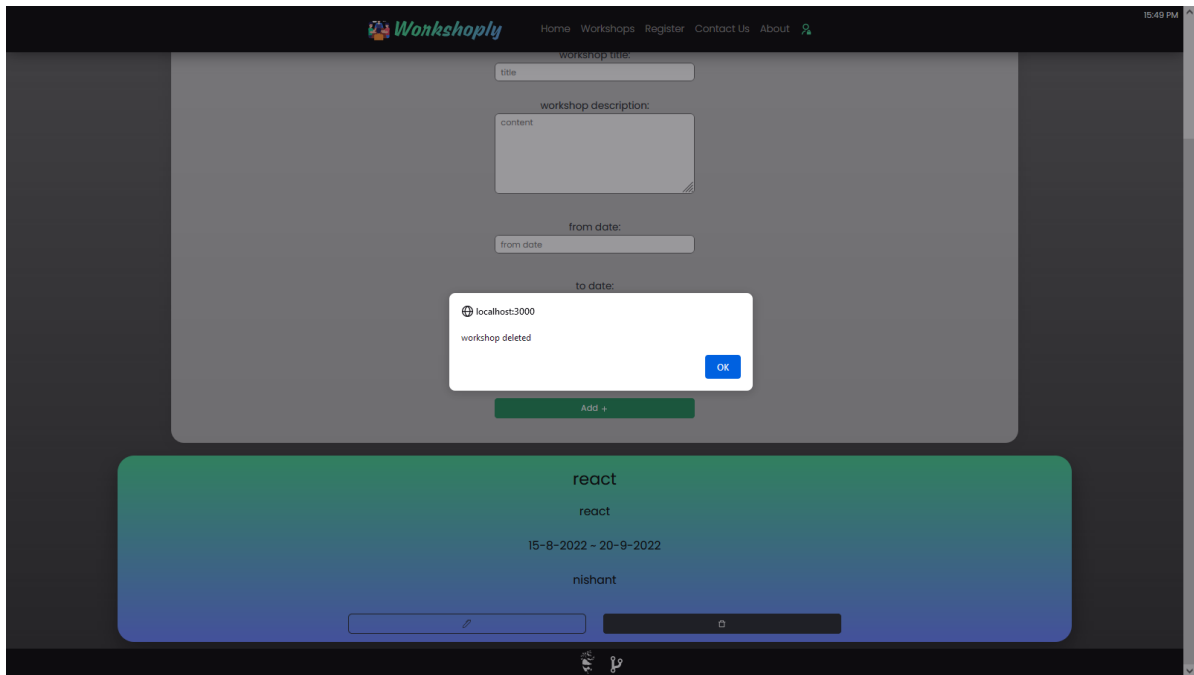
- Updating a particular workshop will only update the workshop with a particular ID in MongoDB Cluster(Atlas), at the same time it will update the workshops at Workshops, and in the dropdown of Register.



Deleting Workshops

- Deleting a workshop will permanently remove all details of the particular workshop in MongoDB Cluster(Atlas), at the same time it will remove that workshop from Workshops, and the dropdown of Register.





Workshoply App Attendees CRUD:

- Admin can Insert an attendee for a particular workshop.
- Users who registered for a workshop are visible to the Admin.
- Admin can Update, Delete and Search the current attendees.

Registration Details



7,698,525,682

React JS

Nishant Jadhav

nisoojadhav@gmail.com

9/10/2022 10:41:39 AM

COMPOSE EMAIL (EMAIL)



1,231,231,231

Calligraphy Workshop

Elon Musk

elonmusk@gmail.com

9/10/2022 11:33:38 AM

COMPOSE EMAIL (EMAIL)



4,564,564,564

Intro to Ember.JS

Warren Buffett

warrenbuffet@yahoo.com

9/16/2022 5:28:01 AM

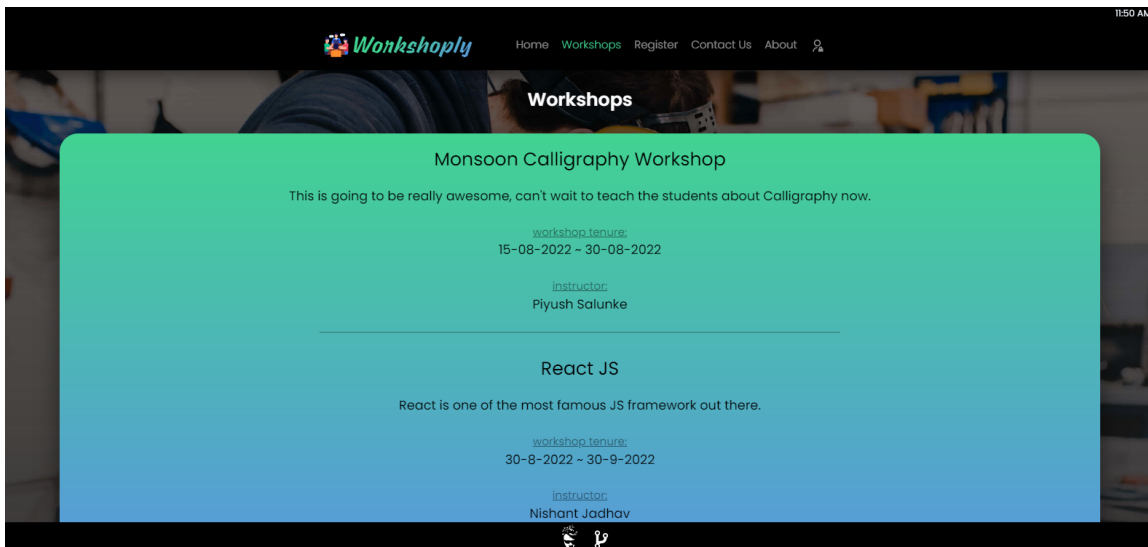
COMPOSE EMAIL (EMAIL)



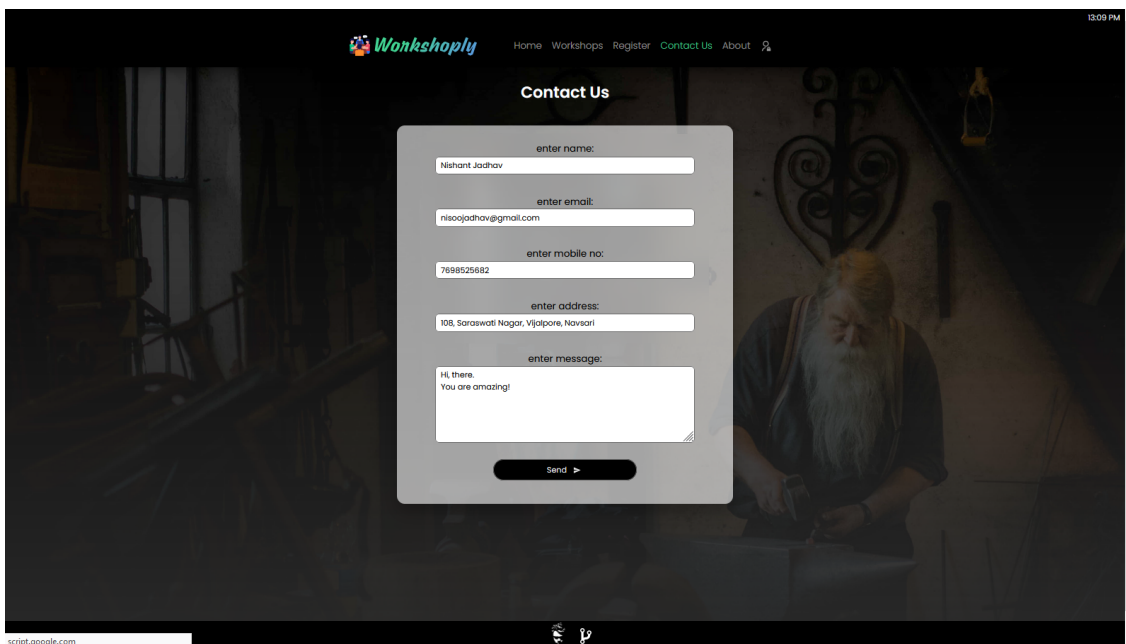
Registration Details

4.2 Other Users(Read(Search), Contact Us, Register)

Reading Workshops



Contact



Contact Responses

Workshop Contact Reponses ☆ 📧 ☁

File Edit View Insert Format Data Tools Extensions Help Last edit was seconds ago

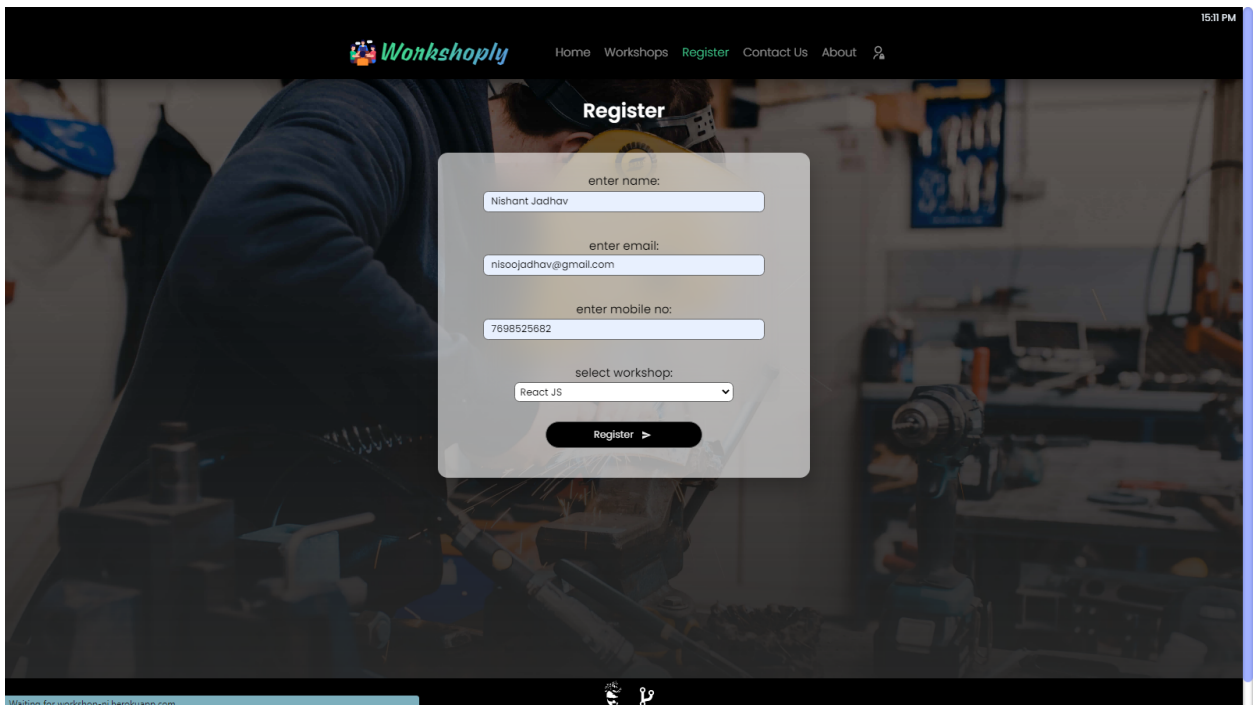
100% £ % .00 .00 123 Arial 10 B I U A

A13:G13 fx 10/09/2022 08:39:01

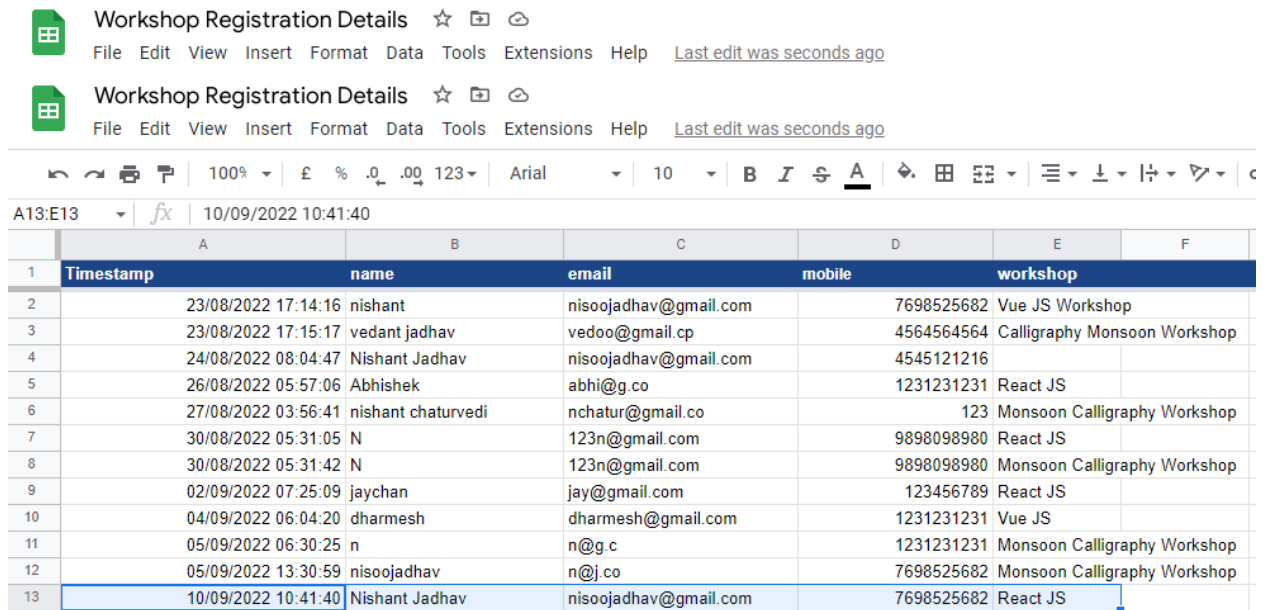
| Timestamp | name | email | mobile | address | message |
|---------------------|----------------|-----------------------|------------|--|-----------------------------|
| 15/07/2022 09:46:13 | ninja | ninja@gmail.com | 123123 | njndjfdjfdhjf hjdh fjhdjfh | Hi there |
| 15/07/2022 09:53:56 | nsii | dfs@dasd.com | 4545121216 | dsdsdsddd | you're awesome |
| 15/07/2022 10:08:44 | dsdsdsds | dds@dds.vs | 123123 | njndjfdjfdhjf hjdh fjhdjfh | final may be |
| 18/07/2022 09:33:41 | Nishant Jadhav | nisoojadhav@gmail.com | 4545121216 | 12120ghf fd dfd fdd fd | Hi there |
| 24/07/2022 06:36:50 | Piyush | pss@g.co | 1111111111 | pune | hi nisoo! |
| 27/07/2022 12:03:50 | vedant | v@gmail.com | 7979797979 | saraswati nagar | hi there |
| 16/08/2022 06:46:29 | vedant | vedoo@gmail.com | 1231231231 | saraswati nagar | hi |
| 23/08/2022 15:02:02 | nishant | nisoojadhav@gmail.com | 7698525682 | 108, Saraswati Nagar, Vijalpore, Navsari | hi there |
| 24/08/2022 08:06:51 | Nishant Jadhav | nisoojadhav@gmail.com | 4545121216 | saraswati nagar | hi |
| 27/08/2022 03:56:08 | nishant | nisoo@gmail.com | 123 | 108, Saraswati Nagar | hi there, nice to meet you |
| 05/09/2022 06:05:15 | n | n@g.co | 1010101010 | | 108 hi |
| 10/09/2022 08:39:01 | Nishant Jadhav | nisoojadhav@gmail.com | 7698525682 | 108, Saraswati Nagar, Vijalpore, Navsari | Hi, there. You are amazing! |

Contact form responses are saved asynchronously on Google Sheets, and mail notification is sent to the admin.

Register



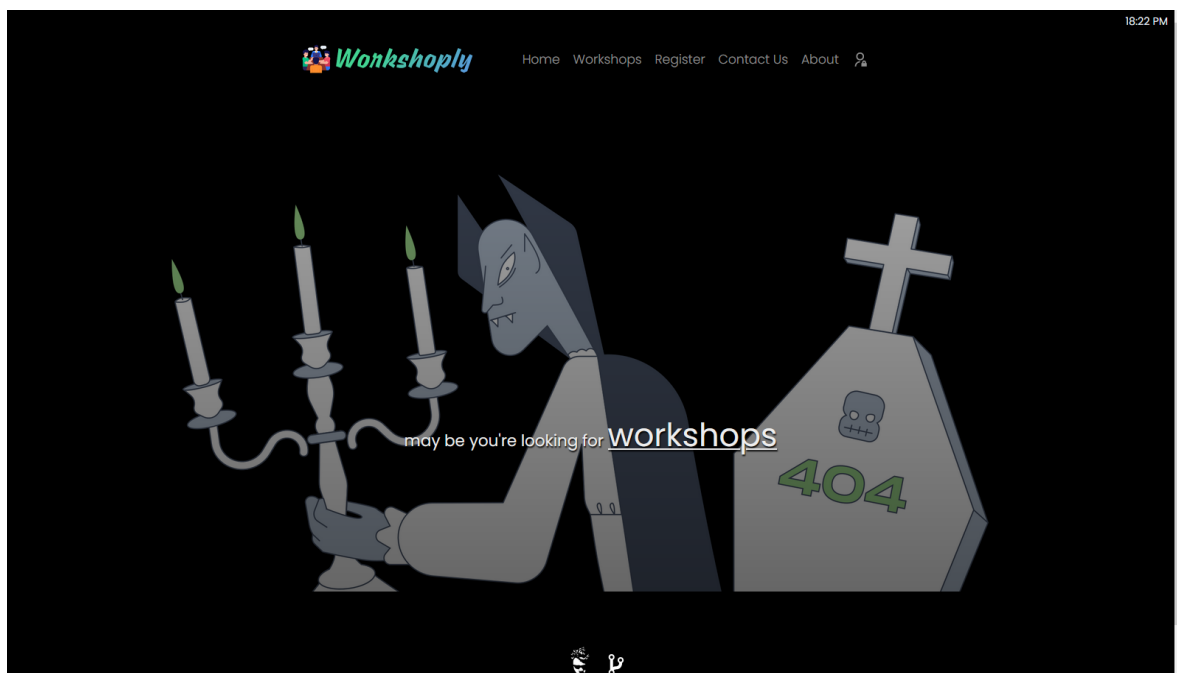
Register Responses



| 1 | Timestamp | name | email | mobile | workshop |
|----|---------------------|--------------------|-----------------------|------------|------------------------------|
| 2 | 23/08/2022 17:14:16 | nishant | nisoojadhav@gmail.com | 7698525682 | Vue JS Workshop |
| 3 | 23/08/2022 17:15:17 | vedant jadhav | vedoo@gmail.cp | 4564564564 | Calligraphy Monsoon Workshop |
| 4 | 24/08/2022 08:04:47 | Nishant Jadhav | nisoojadhav@gmail.com | 4545121216 | |
| 5 | 26/08/2022 05:57:06 | Abhishek | abhi@g.co | 1231231231 | React JS |
| 6 | 27/08/2022 03:56:41 | nishant chaturvedi | nchatur@gmail.co | 123 | Monsoon Calligraphy Workshop |
| 7 | 30/08/2022 05:31:05 | N | 123n@gmail.com | 9898098980 | React JS |
| 8 | 30/08/2022 05:31:42 | N | 123n@gmail.com | 9898098980 | Monsoon Calligraphy Workshop |
| 9 | 02/09/2022 07:25:09 | jaychan | jay@gmail.com | 123456789 | React JS |
| 10 | 04/09/2022 06:04:20 | dharmesh | dharmesh@gmail.com | 1231231231 | Vue JS |
| 11 | 05/09/2022 06:30:25 | n | n@g.c | 1231231231 | Monsoon Calligraphy Workshop |
| 12 | 05/09/2022 13:30:59 | nisoojadhav | n@j.co | 7698525682 | Monsoon Calligraphy Workshop |
| 13 | 10/09/2022 10:41:40 | Nishant Jadhav | nisoojadhav@gmail.com | 7698525682 | React JS |

the Register form responses are saved asynchronously on Google Sheets, and mail notification is sent to the admin, the data is accessible by admin via Workshoply Mobile App.

404 Page

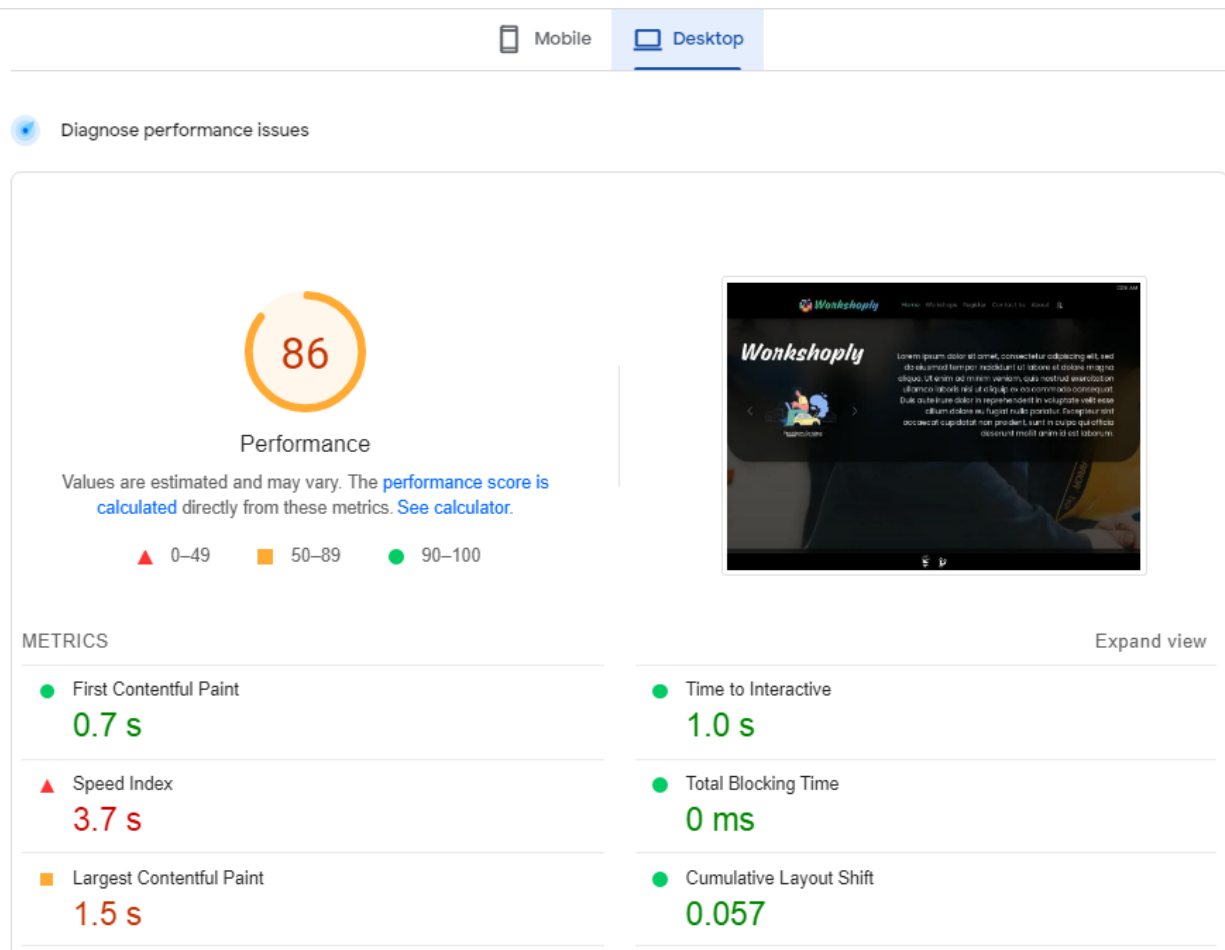


'404 not found' page is displayed to a user when he visits the path that doesn't exist on the the website.

E.g:

<https://workshop-nj.herokuapp.com/hello>

Web Speed Insights



View speed insights on:

https://pagespeed.web.dev/report?url=https%3A%2F%2Fworkshop-nj.herokuapp.com%2F&form_factor=desktop

5 Tools and Technologies

5.1 Overview

Workshoply - is a full stack MERN(MongoDB, Express, React, NodeJS) Application.

Whatever tech stack developed using NodeJS and is currently running on the web requires node modules, node modules are a group of programs packed in one program. One module can depend upon other 1000s of modules, another module can use the module at the same time.

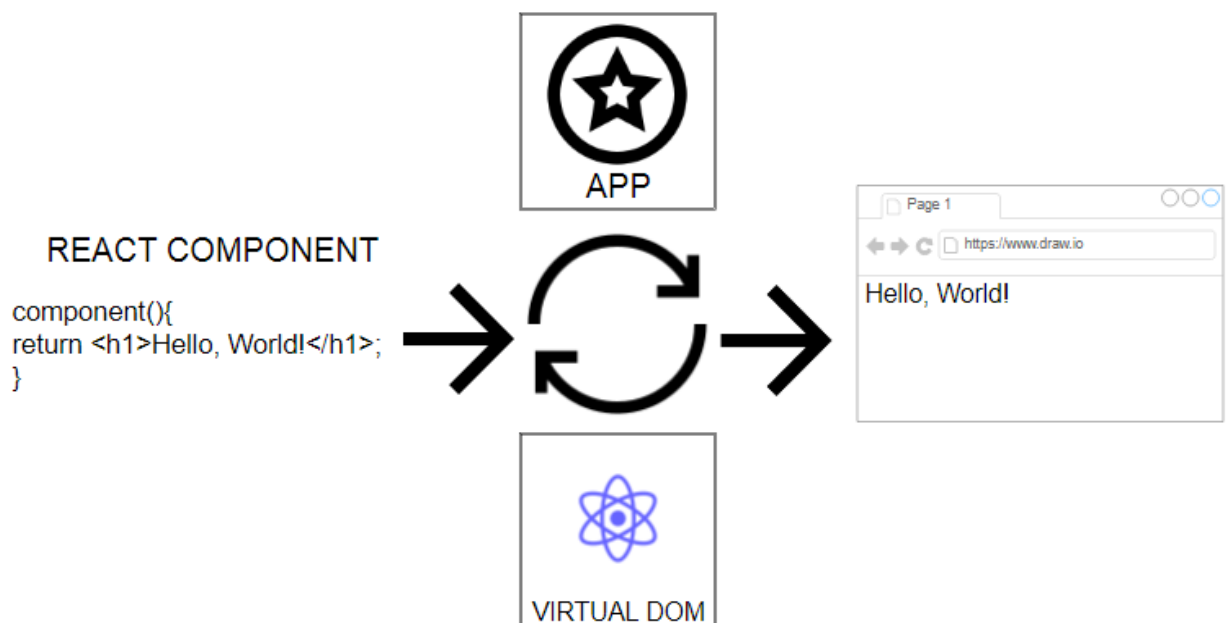
Dependents [\(126\)](#)

tvp-vod-downloader create-r3f-app create-aloudata-app @bahachammakhi/react-cli react-3xd
ac-create-web create-plasmic-app shytools @redtea/generator-react-app visualizer-machines
testofnpm12138 t00ls poyka npm-package-client create-react-wordpress-plugin
generator-reactant yjq-wap @miserylee/create-react-app jsbot-cli oucc-react sandhya123_3
mid-tinker-cli initial-react-app franklin_templeton rigel-app create-ueno-app
create-wptheme-redux yapij @statecharts/xstate-viz presearch-packages
create-wakfu-react-app redux-react-app-cli @graphistry/falcor-react-schema sy-react-init
@jdavalenti/react-cli create-mango-react-app express-react-generator
@nothingmore/generator cli-react generator-lemon-ts @eins78/globals
create-react-with-redux react-workspace-scripts cineasta-cli nickel-front
react-cli-module-structure ratpack create-rrr-app dli yarn-create-react-app
and more...

E.g of dependents from create-react-app npm module:

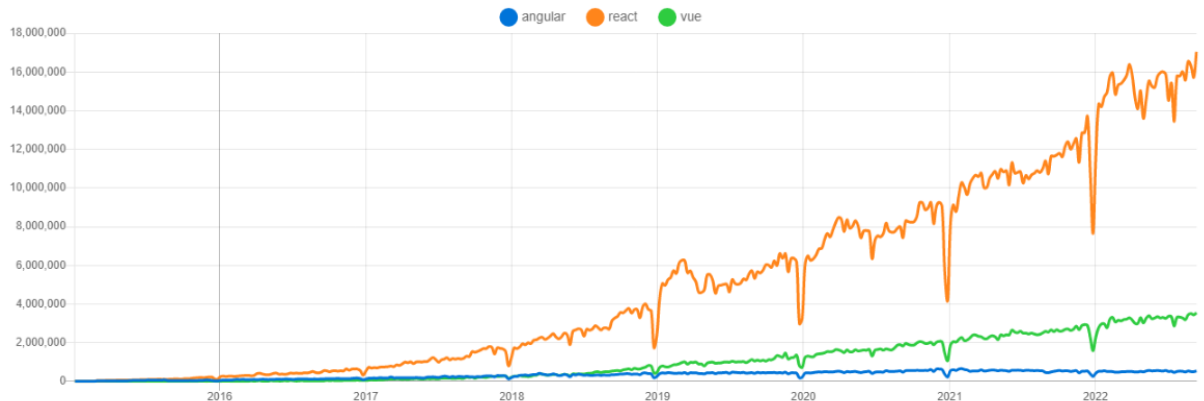
5.2 Why React JS?

- Large developer community
- (With 195k Stars on Github)
- Easy to learn compared to Angular JS
- Efficient debugging
- Virtual DOM
- JSX to increase performance
- Easily develop dynamic websites
- Everything is component
- Reuse of components
- Routing Feature(loads pages with getting refreshed)
- Effective Single Page Application
- Unidirectional data flow
- Open Source



How React Works

Downloads in past All time ▾



Popularity of React all time among Angular, React and Vue

5.3 CSS and Bootstrap

CSS is written from scratch keeping in mind the design rules, typography, font ligature, and responsiveness in mind. Total lines of CSS exceed 800 lines including inline (where it was the need of hour).

Bootstrap is used for modals, react version of Bootstrap: react-bootstrap and react-bootstrap-carousel modules were used to combine with inline JSX.

The complete CSS can be found at:
<https://github.com/NisooJadhav/workshop/blob/main/client/src/App.css>

5.4 Node Modules

Backend:

- **cors**: Cross-Origin Resource Sharing allows a server to indicate any origins (domain, scheme, or port) other than its own from which a browser should permit loading resources.
- **dotenv**: loads environment variables from a .env file into process.env. (to hide MongoDB Atlas connection key).
- **express**: the primary use of Express is to provide server-side logic for web and mobile applications.
- **mongoose**: Mongoose is a MongoDB object modeling tool designed to work in an asynchronous environment. Mongoose supports both promises and callbacks.
- **path**: provides a way of working with directories and file paths.

Frontend:

- **jquery, react-bootstrap**: common frontend styling and functionalities.
- **axios**: to HTTP requests from NodeJS.
- **lodash**: simplify writing arrays, objects, etc.
- **react-bootstrap-carousel**: for carousel in React.
- **react-loader-spinner**: loading spinners.
- **moment**: to parse Date as per our requirement.
- **react-icons**: to add icons in our app.
- **react-dom**: an efficient way of managing DOM elements of web page.

All of the backend and frontend modules can be downloaded using npm or yarn from <https://www.npmjs.com/> or <https://yarnpkg.com/>.

Syntax to install modules:

```
$ npm i module_name
```

5.5 Deployment to Heroku

Prerequisites:

- Working app on local environment.
- Heroku account(<https://heroku.com/>)
- 'Procfile' on root directory of backend code, containing code:

```
web: node server.js
```

- '.env' file on root directory of backend code, containing MongoDB connection key, and password:

```
MONGO_URI="mongodb+srv://username:password  
@cluster0.q4hdt.mongodb.net/workshopsDB"
```

- Add additional lines in server.js:

```
//configuring .env file  
require("dotenv").config();
```

```
//logic for whether the website is local or production  
if(process.env.NODE_ENV === 'production')  
{  
  app.use(express.static("client/build"));  
  app.get("*",(req,res)=>{  
res.sendFile(path.resolve(__dirname,"client","build","index  
.html"));  
  });  
}
```

Directory structure should look like this:

Workshop

|

|->client

 |->build

 |->node_modules

 |->public

 |->src

 package.json

 package-lock.json

|->node_modules

.env

Package.json

Package-lock.json

Procfile

server.js

Build directory inside client directory contains compact production code that is ready to be hosted.

Get into the client directory and run to build production code:

\$ npm run build

PUSHING TO GITHUB:

Create .gitignore file(ignores files specified when pushing to GitHub) in the root directory containing:

.env

node_modules

Get into the root directory and initialize a git repository.

\$ git init

\$ git add .

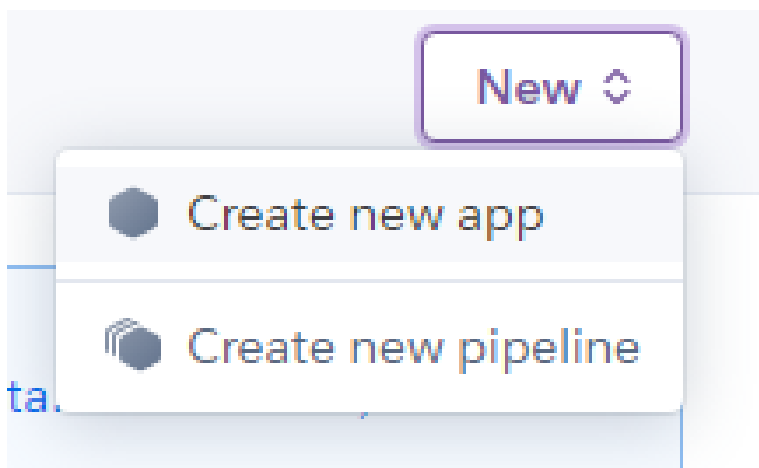
\$ git commit -m "demo commit"

\$ git push

The above four commands will push your project to the GitHub repository.

CREATING PROJECT ON HEROKU:

Visit heroku.com, log in or signup, then select New->Create new app:



Create New App







App name

Choose a region

Add to pipeline...

Create app

Go to Deploy -> Deployment Method, select GitHub to choose your repository -> Connect

| | | |
|--|---|--|
|  Heroku Git Use Heroku CLI |  GitHub Connect to GitHub |  Container Registry Use Heroku CLI |
|  Heroku Git Use Heroku CLI |  GitHub Connect to GitHub |  Container Registry Use Heroku CLI |

Search for a repository to connect to

Missing a GitHub organization? [Ensure Heroku Dashboard has team access.](#)

 NisooJadhav/workshop

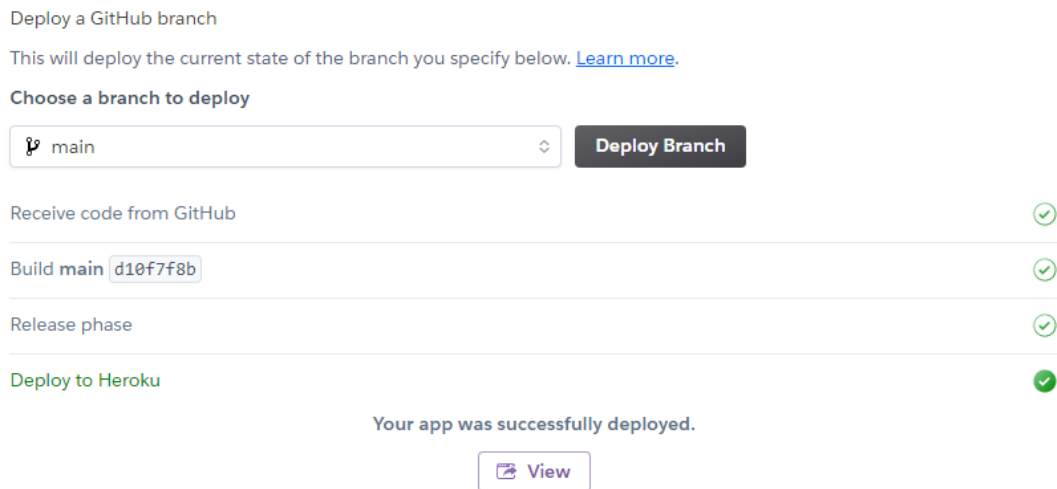
Manual Deploy -> Deploy Branch

Manual deploy
Deploy the current state of a branch to this app.

Deploy a GitHub branch
This will deploy the current state of the branch you specify below. [Learn more.](#)

Choose a branch to deploy

You should success message like this:



The last thing to do, is to add config vars(MongoDB Connection Key, you can get it from MongoDB Atlas Cluster):

Visit settings -> config vars -> reveal config vars -> enter key and value -> add.

5.6 MongoDB Atlas

It's a MongoDB but on the cloud. To get started we can:

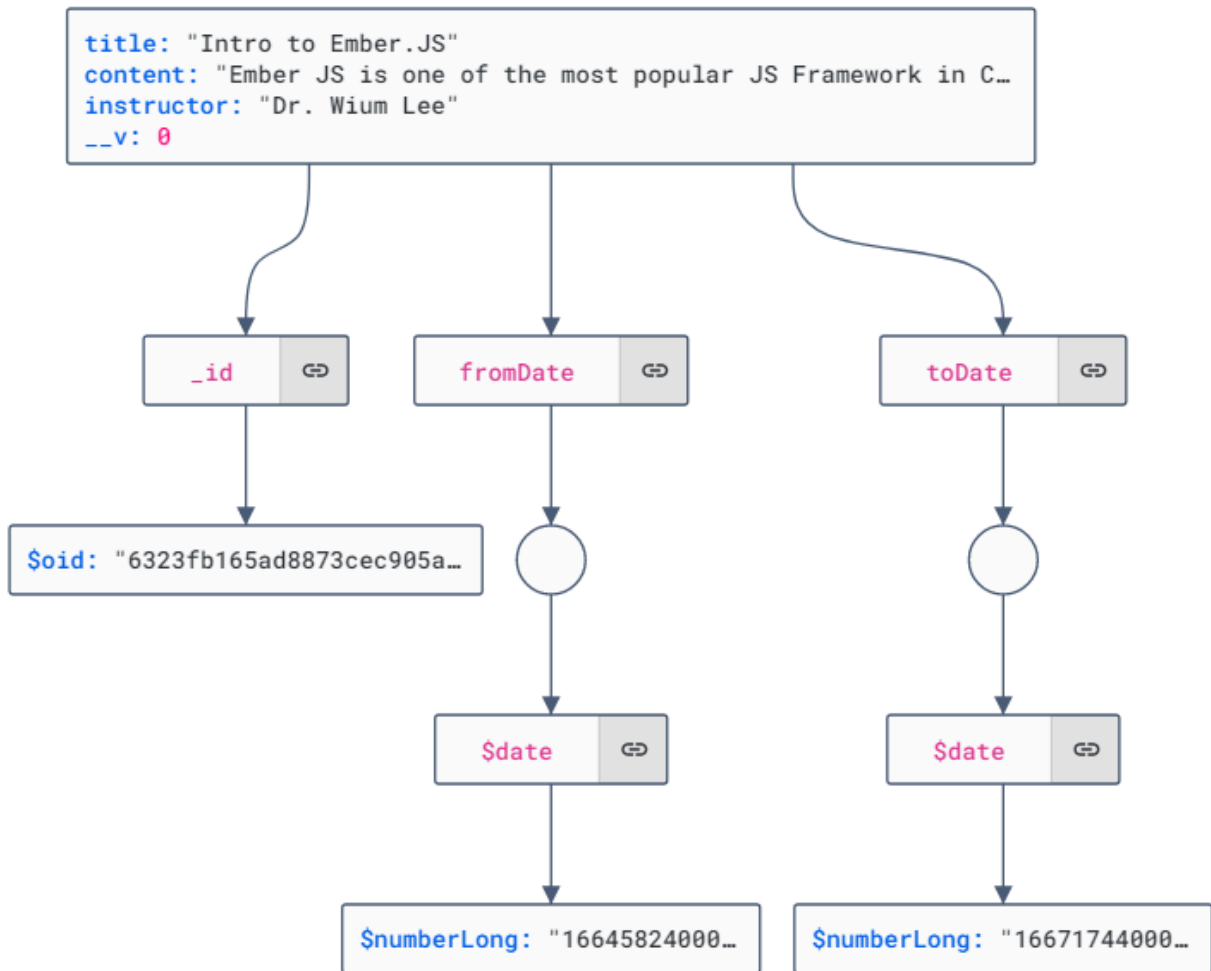
- 1 - visit mongodb.com -> login or signup
- 2 - create cluster -> select location -> free tier.
- 3 - access permission -> from any IP Address.
- 4 - enter password.
- 5 - Copy the connection key, and replace 'password' with your actual password.

5.7 Database Schema

The Document Schema looks like following:

```
{
  "_id": {
    "$oid": "631c67bb3ba8b551d7c45633"
  },
  "title": "Printing Circuit Boards for Raspberry Pi Workshop",
  "content": "Printing Circuit Boards for Raspberry Pi Workshop, I
am really excited to teach you how to print circuit boards.",
  "fromDate": {
    "$date": {
      "$numberLong": "1664150400000"
    }
  },
  "toDate": {
    "$date": {
      "$numberLong": "1666742400000"
    }
  },
  "instructor": "Piyush Salunkhe",
  "__v": 0
}
```

_id is assigned automatically by MongoDB, using hashes. Then we have the title, content, and instructor to be a type of **'string'**. And fromDate and toDate in form of **'date'** respectively.



Visualizing Single Document of Workshop Collection

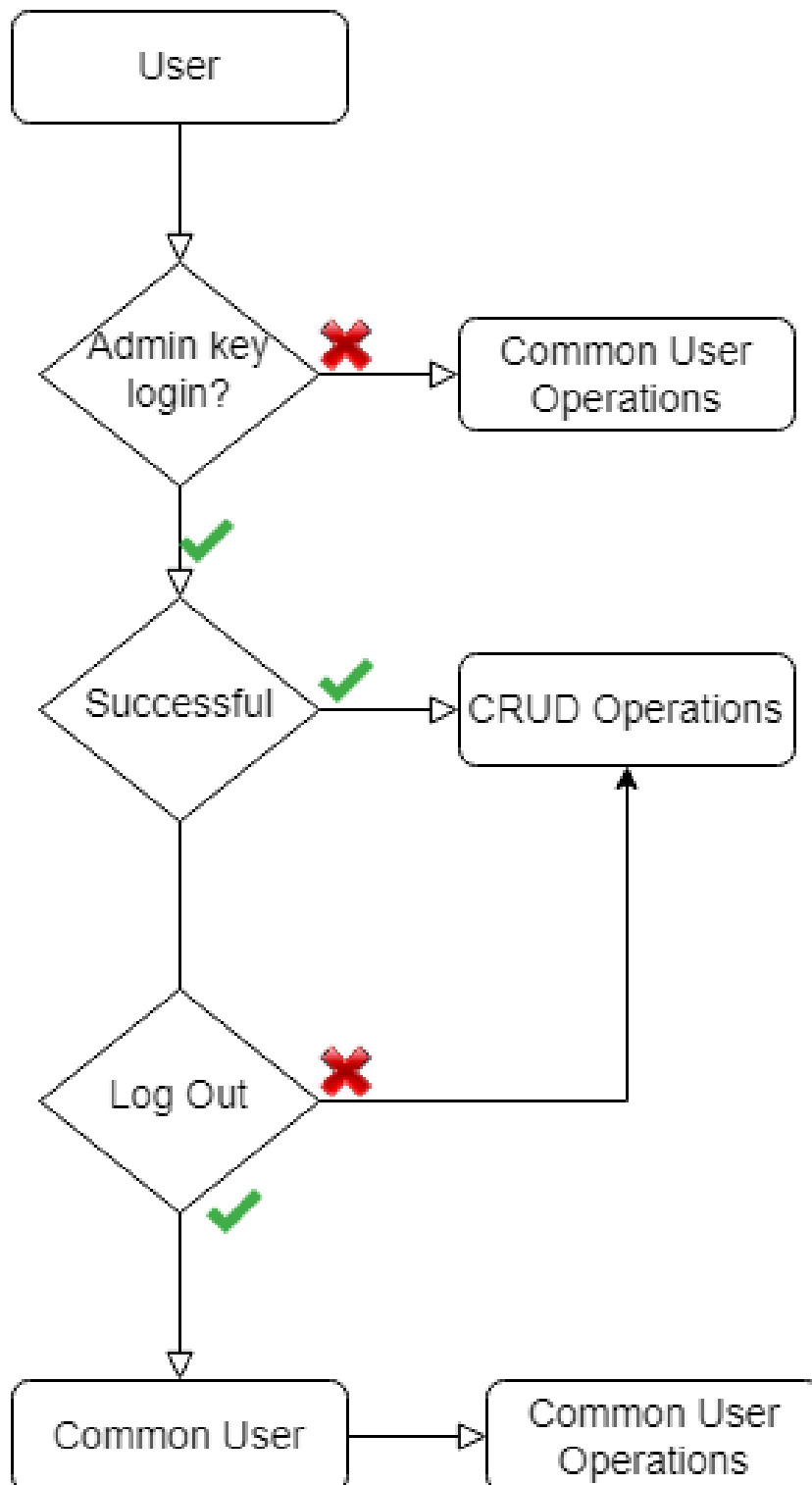
5.8 REST API

Representational State Transfer Application Programming Interface is a tech that uses HTTP requests to access and use data, The data can be used to carry out GET(Read), PUT(Update), POST(Insert), and DELETE(Delete) operations against concerning data resources.

| Method | Route | Description |
|--------|---------------------------|-------------------------------------|
| GET | app.get("/posts") | Get all workshops. |
| PUT | app.put("/update/:id") | Update particular workshop. |
| POST | app.post("/create") | Create new workshop. |
| DELETE | app.delete("/delete/:id") | Delete workshop of a particular id. |

6 System Diagrams

6.1 System Flowchart

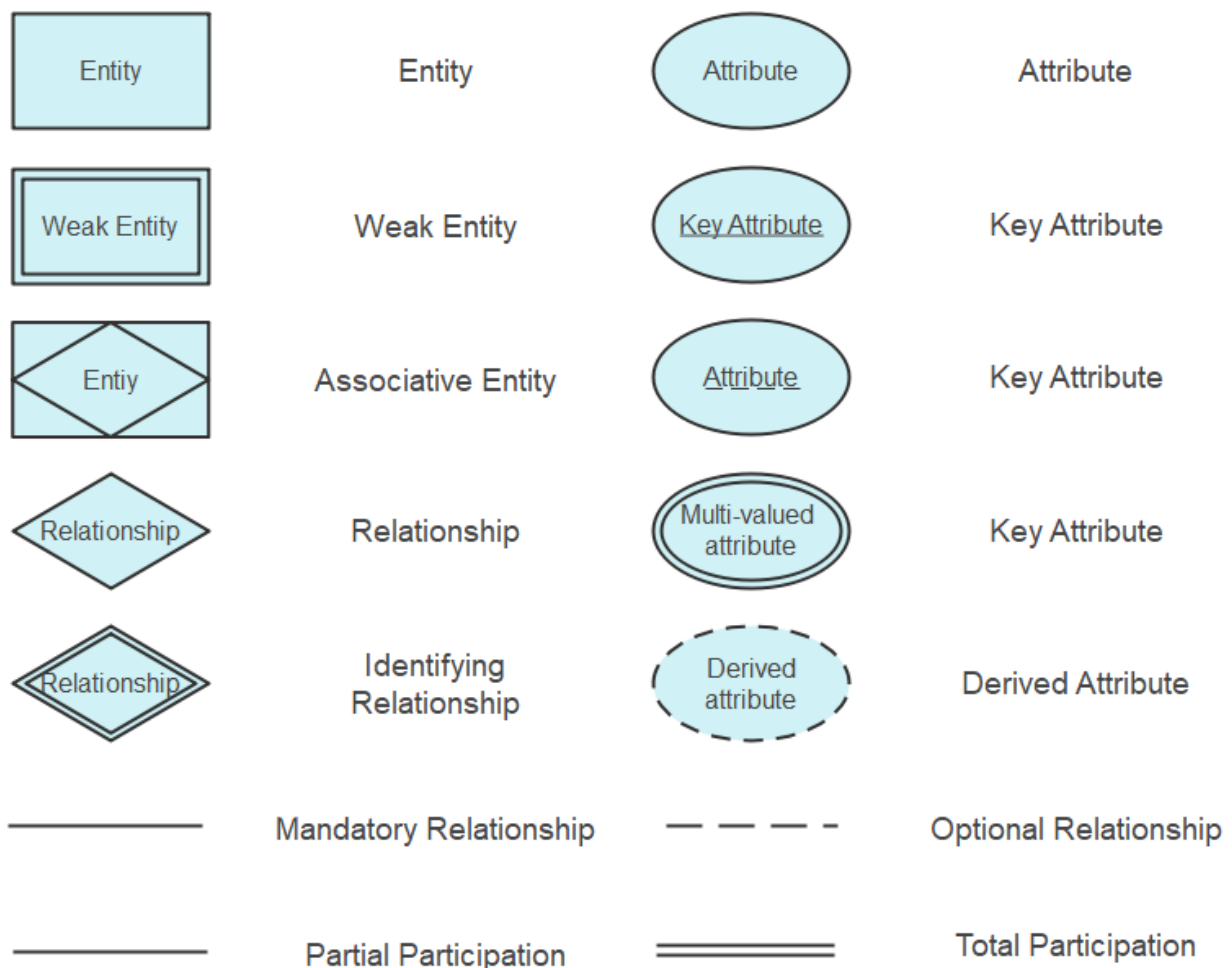


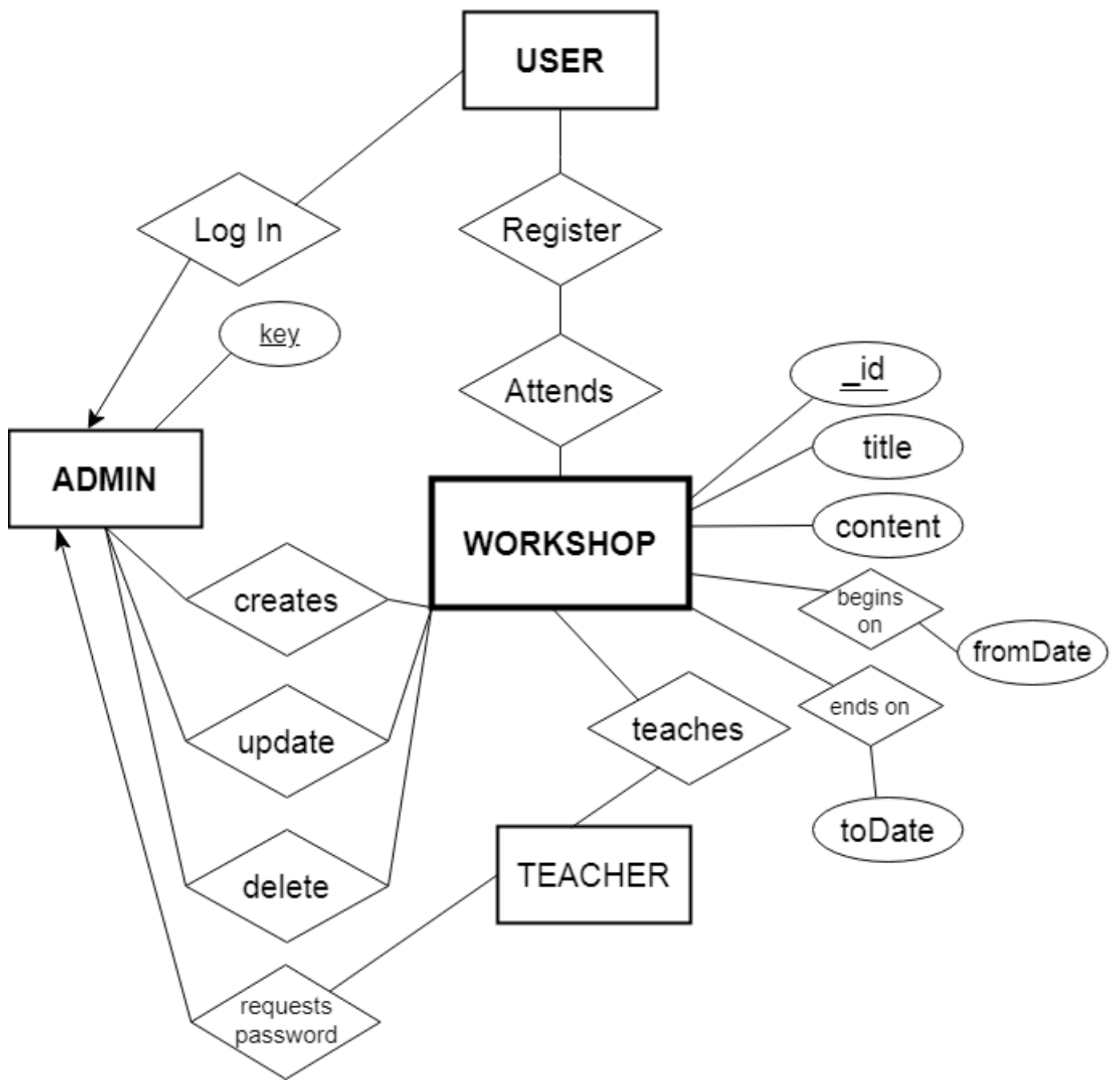
6.2 Entity Relation Diagram

An entity-relationship model (or ER model) describes interrelated things of interest in a specific domain of knowledge.

A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between entities (instances of those entity types).

ER model becomes an abstract data model, that defines a data or information structure that can be implemented in a database, typically a relational database.





Entity Relationship Diagram

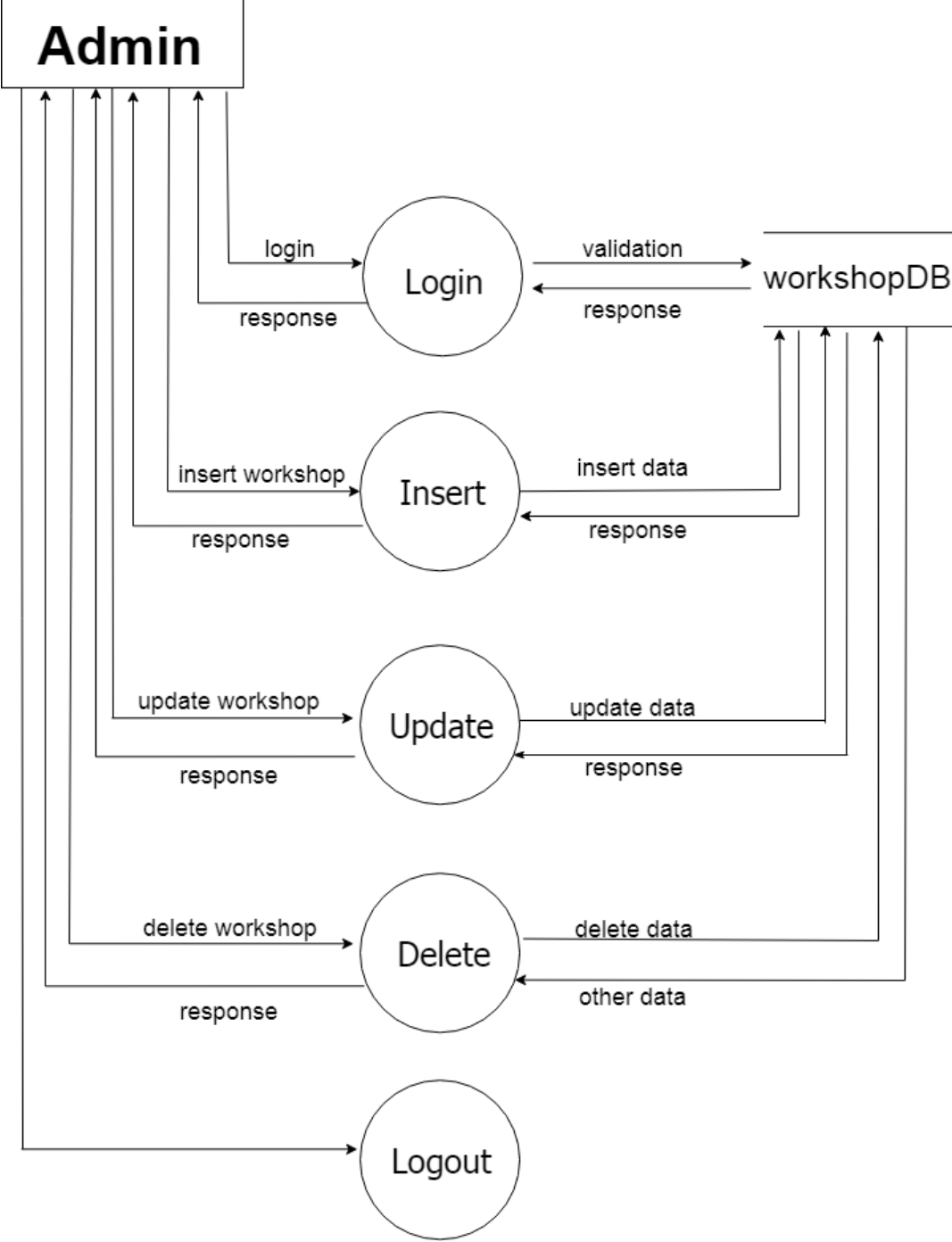
6.3 Data Flow Diagram

- The flow of data of a system or process is represented by DFD. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow – there are no decision rules and no loops.
- For each data flow, at least one of the endpoints (source and/or destination) must exist in a process. The refined representation of a process can be done in another data-flow diagram, which subdivides this process into sub-processes.

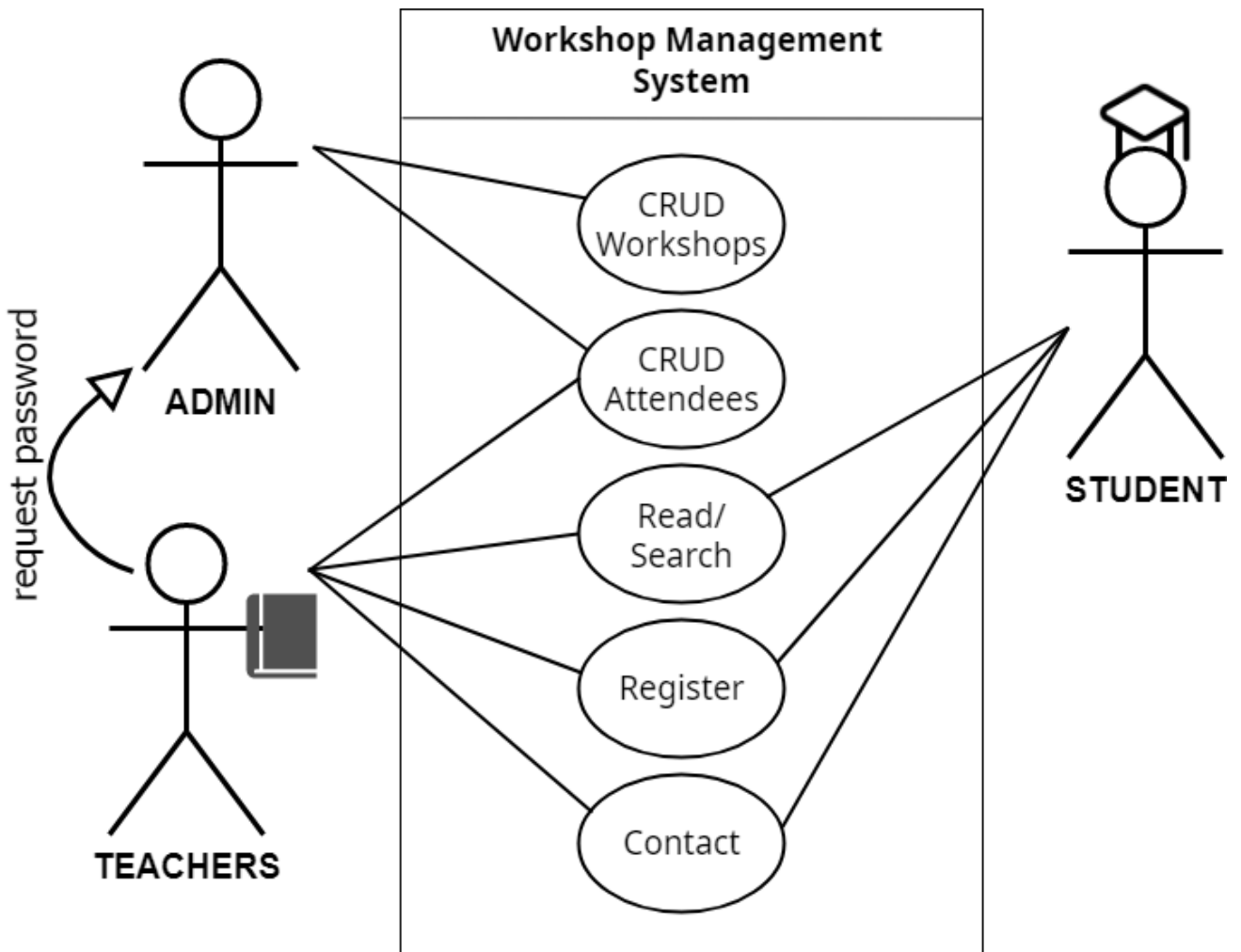
Context Level Diagram:



0 Level Diagram:



6.4 Use Case Diagram



7 Scope of Further Development

New features that can be added to this application:

- New Admin Registration, Log In. The app has only one Admin for now.
- New User Registration, Log In.
- Domain upgradation.
- Adding web notifications.
- Adding paid workshops with a payment interface.
- Generating receipts from booked workshops for a particular user.
- Adding in-built meeting tool, updating schema for inserting images and videos.

The above features will be added by me as I learn and develop new skills, and add them to this application as time permits.

Stay tuned for updates at:

<https://github.com/nisoojadhav/workshop>

8 Important Links

Dynamic Interactive version of this documentation:

<https://workshop-docs.vercel.app/>

Download PDF version of the documentation:

<https://workshop-docs.vercel.app/extra/pdf>

See Video Demonstration:

<https://workshop-docs.vercel.app/extra/video>

See the working version of 'Workshoply':

<http://workshop-nj.herokuapp.com/>

(if the link breaks, visit github.com/nisoojadhav/workshop for updates)

Download the Source Code:

<https://github.com/nisoojadhav/workshop>

9 References

JavaScript - <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

React Basics - https://fullstackopen.com/en/part1/introduction_to_react

React - <https://reactjs.org/docs/getting-started.html>

React-icons - <https://react-icons.github.io/react-icons/>

React Routing -

<https://reactjs.org/docs/code-splitting.html#route-based-code-splitting>

React-Bootstrap - <https://react-bootstrap.github.io/>

Bootstrap - <https://getbootstrap.com/>

Npm - <https://www.npmjs.com/>

MongoDB - <https://www.mongodb.com/>

MongoDB Docs - <https://www.mongodb.com/>

Express - <https://expressjs.com/>

Node JS - <https://nodejs.org/>

YouTube - <https://youtu.be/gotbRizPobA>

StackOverflow - <https://stackoverflow.com/questions/tagged/reactjs>