# Web App Design with React Final Project

**LisaSmith** — 2022-10-05-fesd-nashua — originally due: 3/5/22

***With a 3 week extension, now due: 3/26/22 — Thank you Tyler!***

Points possible: 70

| Category | Criteria | % of Grade |
|---|---|---|
| **Functionality** | Does the code work? | 25 |
| **Organization** | Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear. | 25 |
| **Creativity** | Student solved the problems presented in the assignment using creativity and out of the box thinking. | 25 |
| **Completeness** | All requirements of the assignment are complete. | 25 |

**Instructions:** In VS Code, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your JavaScript project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

**Coding Steps:**

1. Using an online API of your choice, or multiple APIs (or if no API can be found, using an array for in-memory storage is okay as well), create a React project of your choice. You will be working on this for the next three weeks.
2. Project must meet the following criteria:
    a. Use React Router and have at least 3 pages
    b. Use React Bootstrap or an alternative styling library
    c. Contain at least 10 components
    d. Allow for all CRUD operations

**Description*:***

*My Web API called Find-A-Gig, is a working API, which connects Musicians to Available Gigs, and Gig Organizers to Available Musicians. When I started the Front End Coding Bootcamp, my intention was always to connect a front end to that back end. At the moment, my Front End React Final Project — Find-A-Gig-App is a work in progress. I have completed the following:*

1. ***The API used** was crudcrud with the following endpoints:*
    - *https://crudcrud.com/api/467446f2b41f4ea58e1a553fd9faaf72/gigs*
    - *https://crudcrud.com/api/467446f2b41f4ea58e1a553fd9faaf72/users*
    - *https://crudcrud.com/api/467446f2b41f4ea58e1a553fd9faaf72/instruments*
2. ***My React Project, Find-A-Gig-App**:*
    - ***4 pages**: Home, Users, Gigs & Instruments*
    - ***Uses:** React Router v6 & the React Bootstrap library*
    - ***Contains 12 components**: gigs, home, instruments, modal, mybutton, navLinks, newGigForm, newGigInstrumentForm, newInstrumentForm, newUserForm, newUserInstrumentForm, users.*
    - ***Implements all 4 CRUD Operations:** Create, Read, Update, & Delete*
    - ***All API calls are in**: GigApi, InstrumentApi, & UserApi*

**URL to GitHub Repository:** https://github.com/sw-dev-lisa-s-nh/React-Final-Project

**Screenshots of Code:**

**index.js:**

find-a-gig-app > src > JS index.js

```
1   import React from 'react';
2   import ReactDOM from 'react-dom';
3   import 'bootstrap/dist/css/bootstrap.min.css';
4   import './index.css';
5   import { BrowserRouter } from 'react-router-dom';
6   import App from './App';
7   import reportWebVitals from './reportWebVitals';
8
9   ReactDOM.render(
10    <BrowserRouter>
11      <App />
12    </BrowserRouter>,
13    document.getElementById('root')
14  );
15
16  reportWebVitals();
17
```

**App.js:**

find-a-gig-app > src > JS App.js > default

```
import React from 'react';
import '../node_modules/bootstrap/dist/css/bootstrap.min.css';
import Gigs  from './components/gigs';
import Users  from './components/users';
import Instruments  from './components/instruments';
import Navigation from './components/navLinks';
import HomePage from './components/home';
import { Routes, Route } from 'react-router-dom';


function App() {
  return (
    <div>
      <Navigation />
      <Routes>
        <Route path="/" element={ <HomePage />} />
        <Route path="/users" element={ <Users />} />
        <Route path="/instruments" element={ <Instruments />} />
        <Route path="/gigs" element={ <Gigs />} />
      </Routes>
    </div>
  );
}

export default App;
```

**index.css:**

ind-a-gig-app > src > # index.css > h1

```
body {
  margin: 2;
  color: rgb(107, 107, 206);
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue', 'Fantasy',
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  background-image: url('./music.jpg');
}


code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
    monospace;
}

h1,h2,h3,h4,h5,h6, ul, th, td, table,  .accord-custom {
  padding:  2%;
  margin: 2%;
  font-family: fantasy;
  font-display: rgb(107, 107, 206);;
  color: rgb(107, 107, 206);
  background:  rgb(174, 174, 230);
  border: rgb(107, 107, 206);
  box-shadow: 5px 5px 5px rgb(107, 107, 206);

}
```

PROMINEO TECH

**index.css continued:**

nd-a-gig-app > src > # index.css > .new-user

```
.navbar-class, .nav-link, .nav-bar-title, .my-button {
  padding:  2%;
  margin: 2%;
  size: 36;
  text-emphasis-color: rgb(107, 107, 206);
  font-family: fantasy;
  font-display: rgb(107, 107, 206);;
  color: rgb(107, 107, 206);
  background: rgb(174, 174, 230);
  border: rgb(107, 107, 206);
  box-shadow: 5px 5px 5px rgb(107, 107, 206);
}


.new-user,  .new-card, .new-instrument, .info-button, .instrument-list{
  margin: 2%;
  padding:  1%;
  font-family: fantasy;
  align-content: space-between;
  color: rgb(66, 66, 97);
  background: rgb(198, 198, 207);
  border-radius: 4px;
  border-color: rgb(107, 107, 206);
  border-color: rgb(107, 107, 206);
  box-shadow: 5px 5px 5px rgb(107, 107, 206);
}

.new-form {
  font-family: fantasy;
  color: rgb(34, 34, 202);
  border-color: rgb(107, 107, 206);
  border-color: rgb(107, 107, 206);
  box-shadow: 5px 5px 5px rgb(107, 107, 206);
}


.button-custom {
  size: lg;
  size: 36px;
  border-color: rgb(107, 107, 206);
  box-shadow: 5px 5px 5px rgb(107, 107, 206);
}


ul {
  font-size:  24px;
}

.gig-card {
  background-color: rgb(107, 107, 206);
}

.info-gig-card {
  background-color: rgb(209, 209, 250);
}
```

PROMINEO TECH

**GigApi.js:**

nd-a-gig-app > src > rest > JS GigApi.js > ...

```
const GIG_ENDPOINT = 'https://crudcrud.com/api/467446f2b41f4ea58e1a553fd9faaf72/gigs';

class GigApi {
    get = async () => {
        try {
            console.log(GIG_ENDPOINT)
            const resp = await fetch(`${GIG_ENDPOINT}`);
            console.log(resp);
            const data = await resp.json();
            console.log(data);
            return data;
        } catch(e) {
            console.log('Error:  get gigs had an issue.', e);
        }

    }

    getOne = async (id) => {
        try {
            console.log(GIG_ENDPOINT)
            const resp = await fetch(`${GIG_ENDPOINT}/${id}`);
            console.log(resp);
            const data = await resp.json();
            console.log(data);
            return data;
        } catch(e) {
            console.log(`Error:  get gig id: ${id} had an issue.`, e);
        }

    }

    post = async (gig) => {
        try {
            const resp = await fetch(`${GIG_ENDPOINT}`, {
                method: 'POST',
                headers: {
                    'Content-Type' : 'application/json'
                },
                body: JSON.stringify(gig)
            });
            return await resp.json();
        } catch(e) {
            console.log('Error:  post gig had an issue.', e);
        }
    }

    put = async (gig) => {
        try {
            console.log(GIG_ENDPOINT);
            const resp = await fetch(`${GIG_ENDPOINT}/${gig._id}`, {
                method: 'PUT',
                headers: {
                    'Content-Type' : 'application/json'
                },
                body: JSON.stringify({"name" : gig.name, "date" : gig.date, "address": gig.address,
                "phone" : gig.phone, "description" : gig.description, "instruments" : gig.instruments}),
            });
            console.log(resp);
            return await resp;
        } catch(e) {
            console.log('Error:  update gig had an issue.', e);
        }
    }

    delete = async (gigId) => {
        try {
            console.log(GIG_ENDPOINT);
            const resp = await fetch(`${GIG_ENDPOINT}/${gigId}`, {
                method: 'DELETE'
            });
            console.log(resp);
        } catch(e) {
            console.log('Error:  delete gig had an issue.',e);
        }
    }
}

// create an instance of this class, and we can import that instance.
export const gigApi = new GigApi();
```

PROMINEO TECH

InstrumentApi.js:

find-a-gig-app > src > rest > JS UserApi.js > ...

```
const USER_ENDPOINT = 'https://crudcrud.com/api/467446f2b41f4ea58e1a553fd9faaf72/users';

class UserApi {
    get = async () => {
        try {
            console.log(`${USER_ENDPOINT}`)
            const resp = await fetch(`${USER_ENDPOINT}`);
            console.log(resp);
            const data = await resp.json();
            console.log(data);
            return data;
        } catch(e) {
            console.log('Error:  get users had an issue.', e);
        }

    }

    getOne = async (id) => {
        try {
            console.log(`${USER_ENDPOINT}`)
            const resp = await fetch(`${USER_ENDPOINT}/${id}`);
            console.log(resp);
            const data = await resp.json();
            console.log(data);
            return data;
        } catch(e) {
            console.log(`Error:  get user id: ${id} had an issue.`, e);
        }

    }


    post = async (user) => {
        try {
            const resp = await fetch(`${USER_ENDPOINT}`, {
                method: 'POST',
                headers: {
                    'Content-Type' : 'application/json'
                },
                body: JSON.stringify(user)
            });
            return await resp.json();
        } catch(e) {
            console.log('Error: post users had an issue.', e);
        }
    }

    put = async (user) => {
        try {
            console.log(USER_ENDPOINT);
            const resp = await fetch(`${USER_ENDPOINT}/${user._id}`, {
                method: 'PUT',
                headers: {
                    'Content-Type' : 'application/json'
                },
                body: JSON.stringify({
                "firstname" : user.firstname,
                "lastname" : user.lastname,
                "address" : user.address,
                "city" : user.city,
                "state" : user.state,
                "phone" : user.phone,
                "instruments": user.instruments }),
            });
            console.log(resp);
            //return resp.json();
            return resp;
        } catch(e) {
            console.log('Error:  update users had an issue.', e);
        }
    }

    delete = async (userId) => {
        try {
            console.log(USER_ENDPOINT);
            const resp = await fetch(`${USER_ENDPOINT}/${userId}`, {
                method: 'DELETE'
            });
            console.log(resp);
        } catch(e) {
            console.log('Error:  delete user had an issue.',e);
        }
    }
}

// create an instance of this class, and we can import that instance.
export const userApi = new UserApi();

```

# PROMINEO TECH

UserApi.js:

find-a-gig-app > src > rest > JS UserApi.js > ...

```
const USER_ENDPOINT = 'https://crudcrud.com/api/467446f2b41f4ea58e1a553fd9faaf72/users';

class UserApi {
    get = async () => {
        try {
            console.log(`${USER_ENDPOINT}`)
            const resp = await fetch(`${USER_ENDPOINT}`);
            console.log(resp);
            const data = await resp.json();
            console.log(data);
            return data;
        } catch(e) {
            console.log('Error:  get users had an issue.', e);
        }

    }

    getOne = async (id) => {
        try {
            console.log(`${USER_ENDPOINT}`)
            const resp = await fetch(`${USER_ENDPOINT}/${id}`);
            console.log(resp);
            const data = await resp.json();
            console.log(data);
            return data;
        } catch(e) {
            console.log(`Error:  get user id: ${id} had an issue.`, e);
        }

    }


    post = async (user) => {
        try {
            const resp = await fetch(`${USER_ENDPOINT}`, {
                method: 'POST',
                headers: {
                    'Content-Type' : 'application/json'
                },
                body: JSON.stringify(user)
            });
            return await resp.json();
        } catch(e) {
            console.log('Error: post users had an issue.', e);
        }
    }

    put = async (user) => {
        try {
            console.log(USER_ENDPOINT);
            const resp = await fetch(`${USER_ENDPOINT}/${user._id}`, {
                method: 'PUT',
                headers: {
                    'Content-Type' : 'application/json'
                },
                body: JSON.stringify({
                "firstname" : user.firstname,
                "lastname" : user.lastname,
                "address" : user.address,
                "city" : user.city,
                "state" : user.state,
                "phone" : user.phone,
                "instruments": user.instruments }),
            });
            console.log(resp);
            //return resp.json();
            return resp;
        } catch(e) {
            console.log('Error:  update users had an issue.', e);
        }
    }

    delete = async (userId) => {
        try {
            console.log(USER_ENDPOINT);
            const resp = await fetch(`${USER_ENDPOINT}/${userId}`, {
                method: 'DELETE'
            });
            console.log(resp);
        } catch(e) {
            console.log('Error:  delete user had an issue.',e);
        }
    }
}

// create an instance of this class, and we can import that instance.
export const userApi = new UserApi();

```

PROMINEO TECH

**modal.js:**

mybutton.js 1 | instruments.js 1 | modal.js M | gigs.js 1 | users.js 1 | home.js

find-a-gig-app > src > components > modal.js > ModalWindow

```
import React from 'react';
import { useState } from 'react';
import { Modal } from 'react-bootstrap';

function ModalWindow() {
    const [show, setShow] = useState(false);
    const handleClose = () => setShow(false);
    const handleShow = () => setShow(true);

    return(
        <>
            <button onClick={handleShow}>
                Find-A-Gig-App
            </button>

            <Modal show={show} onHide={handleClose}>
                <Modal.Header closeButton>
                    <Modal.Title>Find-A-Gig-App Description</Modal.Title>
                </Modal.Header>
                <Modal.Body>
                    <strong>Find-A-Gig-App</strong> has a two-fold purpose. <br /> <br />  It is a way for
                    <strong>Musicians</strong> to register, listing the instruments that they play; and to
                    request a booking to play at Open Gigs. <br />  <br /> It also allows <strong>Event
                    Planners</strong> to register gigs, listing instruments needed; and to connect and
                    hire musicians!
                </Modal.Body>
                <Modal.Footer>
                    <button  onClick={handleClose}>
                        Close
                    </button>
                </Modal.Footer>
            </Modal>

        </>
    );
}
```

**home.js:**

mybutton.js 1 | instruments.js 1 | gigs.js 1 | users.js 1 | home.js M

find-a-gig-app > src > components > home.js > ...

```
import React from 'react';
import '../../node_modules/bootstrap/dist/css/bootstrap.min.css';
// import { Popover } from 'react-bootstrap';

import MyButton from './mybutton';


function HomePage (props) {
    return (
        <div className="container">
            <div className="row mt-5">
                <div className="col-md-8 mx-auto">
                    <h1 className="text-center">Welcome to Find-A-Gig</h1>
                </div>
            </div>
            <div className="col-md-2 flex mx-auto" >
                <MyButton className="info-button" />
            </div>
        </div>
    )
}

export default HomePage;
```

PROMINEO TECH

**mybutton.js:**

mybutton.js 1 | instruments.js 1 | gigs.js 1 | users.js 1

find-a-gig-app > src > components > mybutton.js > MyButton

```
1   import React  from 'react';
2   import { useState } from 'react';
3   import '../../node_modules/bootstrap/dist/css/bootstrap.min.css';
4   // import { Button } from 'react-bootstrap';
5   import ModalWindow from './modal';
6
7   function MyButton (props) {
8
9       const { label } = props;
10      const [show, setShow] = useState(false);
11
12      return (
13          <div className="container my-3">
14              <button className="info-button"
15                      onClick={() => setShow(true)}
16              >
17              <br />
18              <strong>{label}  </strong>
19                  <ModalWindow />
20              <br /><br />
21              </button>
22          </div>
23      );
24
25  }
26
27  export default MyButton;
28
```

# PROMINEO TECH

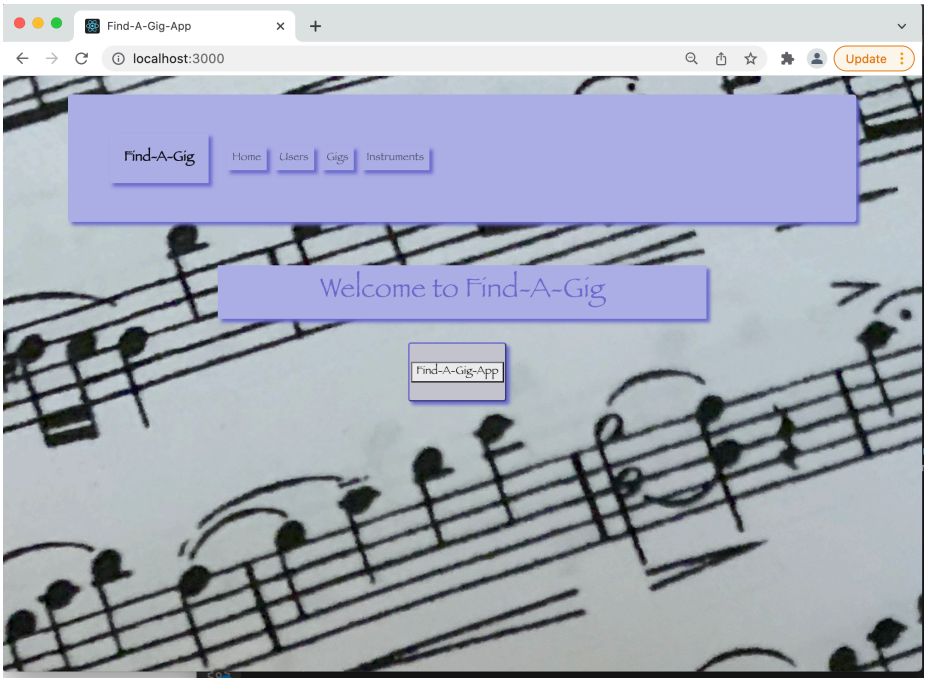
**gigs.js:**

```
find-a-gig-app > src > components > gigs.js > Gigs > useEffect() callback
import React from 'react';
import '../../node_modules/bootstrap/dist/css/bootstrap.min.css';
import { useState, useEffect } from 'react';
import { Card } from 'react-bootstrap'
import  NewGigForm  from './newGigForm';
import { gigApi } from '../rest/GigApi.js';


function Gigs (props) {
    const[error] = useState(null);
    const[isLoaded, setIsLoaded] = useState(false);
    const[gig, setGig] = useState([])
   // const[gigInstruments, setGigInstruments] = useState([]);

    const createGig = async (newGig) => {
        await gigApi.post(newGig);
        runThisEveryTime();
    };

    const deleteGig = async (gigId) => {
        await gigApi.delete(gigId);
        runThisEveryTime();
    };

    const removeGigInstrument = async (gigId, instrumentId)  => {
        console.log("In removeGigInstrument", gigId);
        let newGig = await gigApi.getOne(gigId);
        console.log("In removeGigInstrument, after getOne", newGig);
        const updatedGig =  {
            ...newGig,
            instruments: newGig.instruments.filter((x) => x._id !== instrumentId)
        };
        console.log("updatedGig: " + updatedGig);
        gigApi.put(updatedGig);
        runThisEveryTime();
    }
    const runThisEveryTime  = (() => {
        fetch('https://crudcrud.com/api/467446f2b41f4ea58e1a553fd9faaf72/gigs')
       .then( res => res.json() )
        .then(
            (result) => {
                setIsLoaded(true);
                setGig(result);
                console.log(result);
            },
            (error) => {
                setIsLoaded(true);
                setGig(error);
                console.log(error);
            }
        )
    });

    useEffect( () => {
        runThisEveryTime();
    }, []);

    if (error) {
        return <div>An error was encountered: {error.message}</div>
    } else {
        return(
            <div className="container">
                <div className="row my-4">
                    <div className="col-md-8 mx-auto">
                        <h2 className="text-center"><strong>Available Gigs</strong></h2>
                    </div>
                </div>
                <div className="row">
                    {gig.map( (g, ind) => (
                        <div className="gig-card col-4 my-3" key={g._id}>
                            <Card className="info-gig-card m-2 p-2">
                                <Card.Body>
                                <Card.Title>{ind+1}.   <strong>{g.name}</strong></Card.Title>
                                <Card.Text>
                                    Id: {g._id} <br />
                                    <strong>On: </strong> {g.date} <br />
                                    <strong>Located at:</strong> <br/>
                                    {g.address}, {g.city}, {g.state}.<br /><br />
                                    <strong>Description:</strong><span> </span>{g.description}<br /><br />
                                    <strong>Contact Info: </strong> <span> </span>{g.phone}
                                    <br /><strong>Instruments Requested:</strong>
                                    <br />
                                    <ul>
                                    {g.instruments.map( (inst, j) => (

                                        <li className="border" key={inst._id}>   
                                            <strong>{inst.name} </strong>
                                        <br />
                                        <button className="btn-my-color rounded" onClick={e =>
                                            removeGigInstrument(g._id, inst._id)}>Remove </button>
                                        </li>
                                    ))}
                                    </ul>
                                </Card.Text>
                                </Card.Body>
                                <Card.Footer>
                                <button className="btn-my-color rounded" onClick={e =>
                                    deleteGig(g._id)}>Delete Gig</button>
                                </Card.Footer>
                            </Card>
                        </div>
                    ))}
                </div>
                <NewGigForm createGig={createGig} />
            </div>
        );
    }
}

export default Gigs;
```

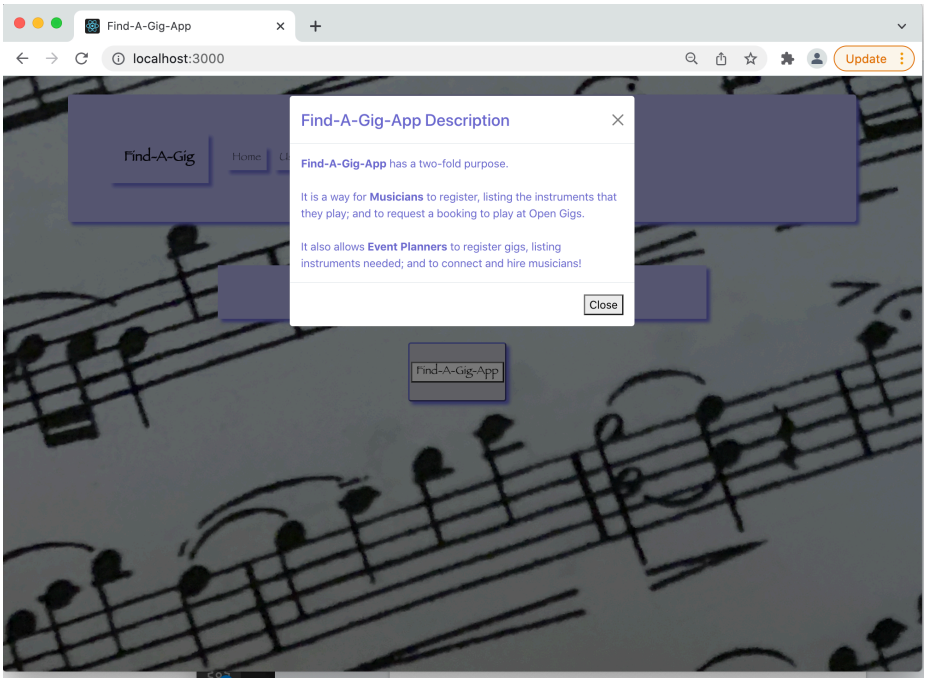
**See the following files on GitHub:** instruments.js, users.js, newGigForm.js, newGigInstrumentForm.js, newInstrumentForm.js, newUserForm.js, newUserInstrumentForm.js

**Screenshots of Running Application:**

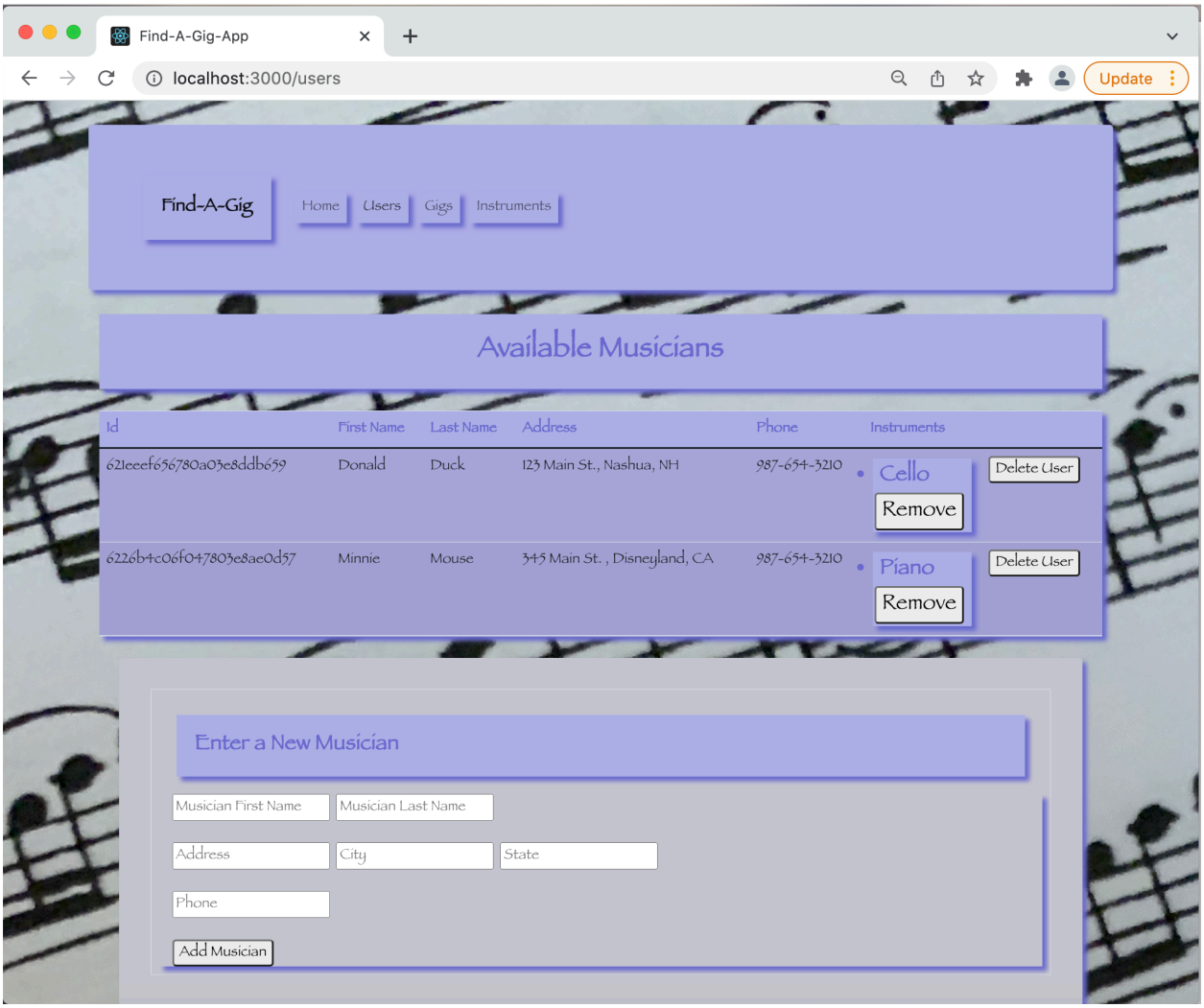
**Home Page:**

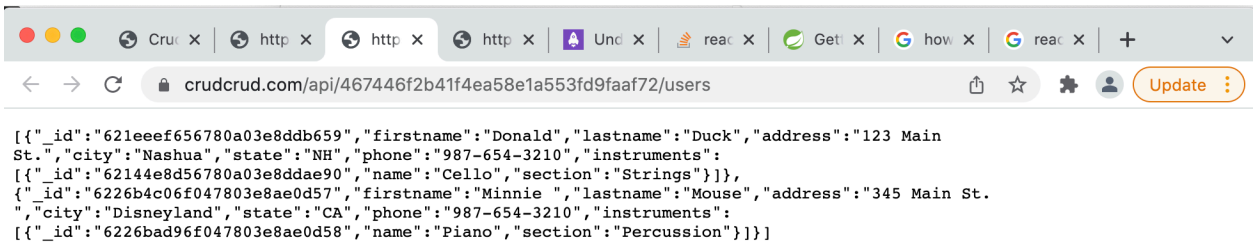



Click on the button (pops up a modal):



**Users Page** (Read)**:**

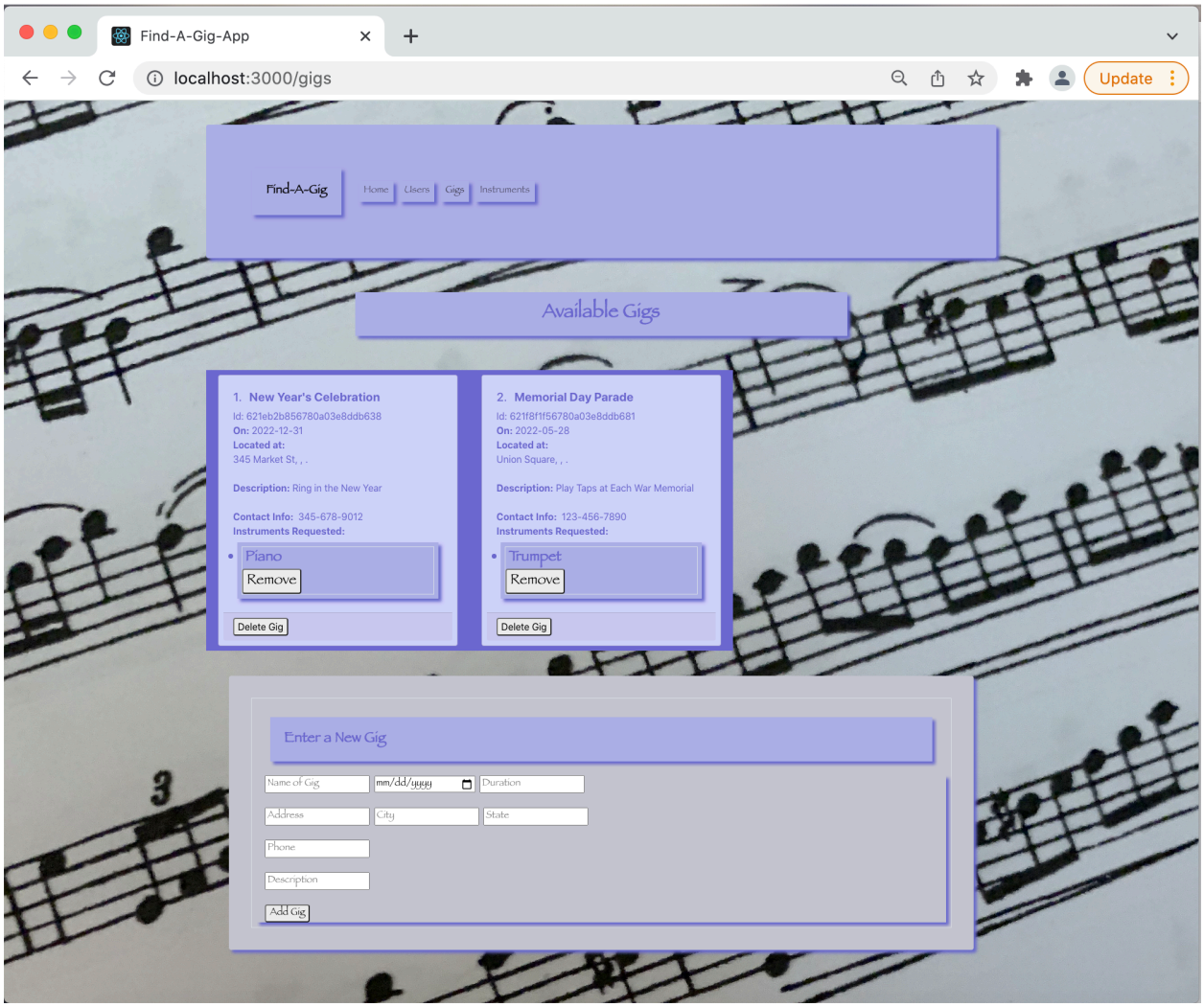


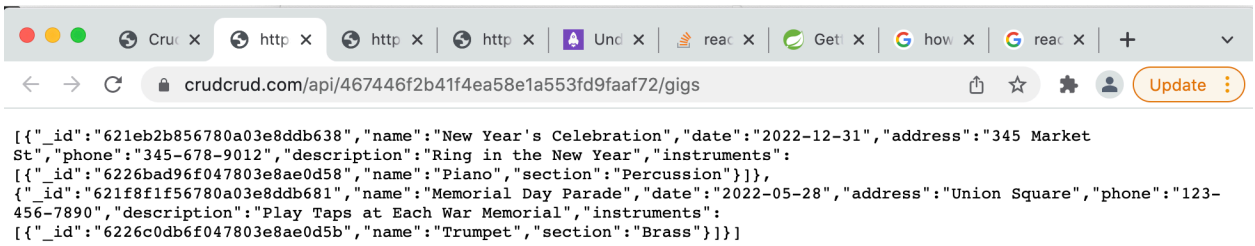


```
[{"_id":"621eeef656780a03e8ddb659","firstname":"Donald","lastname":"Duck","address":"123 Main
St.","city":"Nashua","state":"NH","phone":"987-654-3210","instruments":
[{"_id":"62144e8d56780a03e8ddae90","name":"Cello","section":"Strings"}]},
{"_id":"6226b4c06f047803e8ae0d57","firstname":"Minnie ","lastname":"Mouse","address":"345 Main St.
","city":"Disneyland","state":"CA","phone":"987-654-3210","instruments":
[{"_id":"6226bad96f047803e8ae0d58","name":"Piano","section":"Percussion"}]}]
```

Gigs Page (Read):

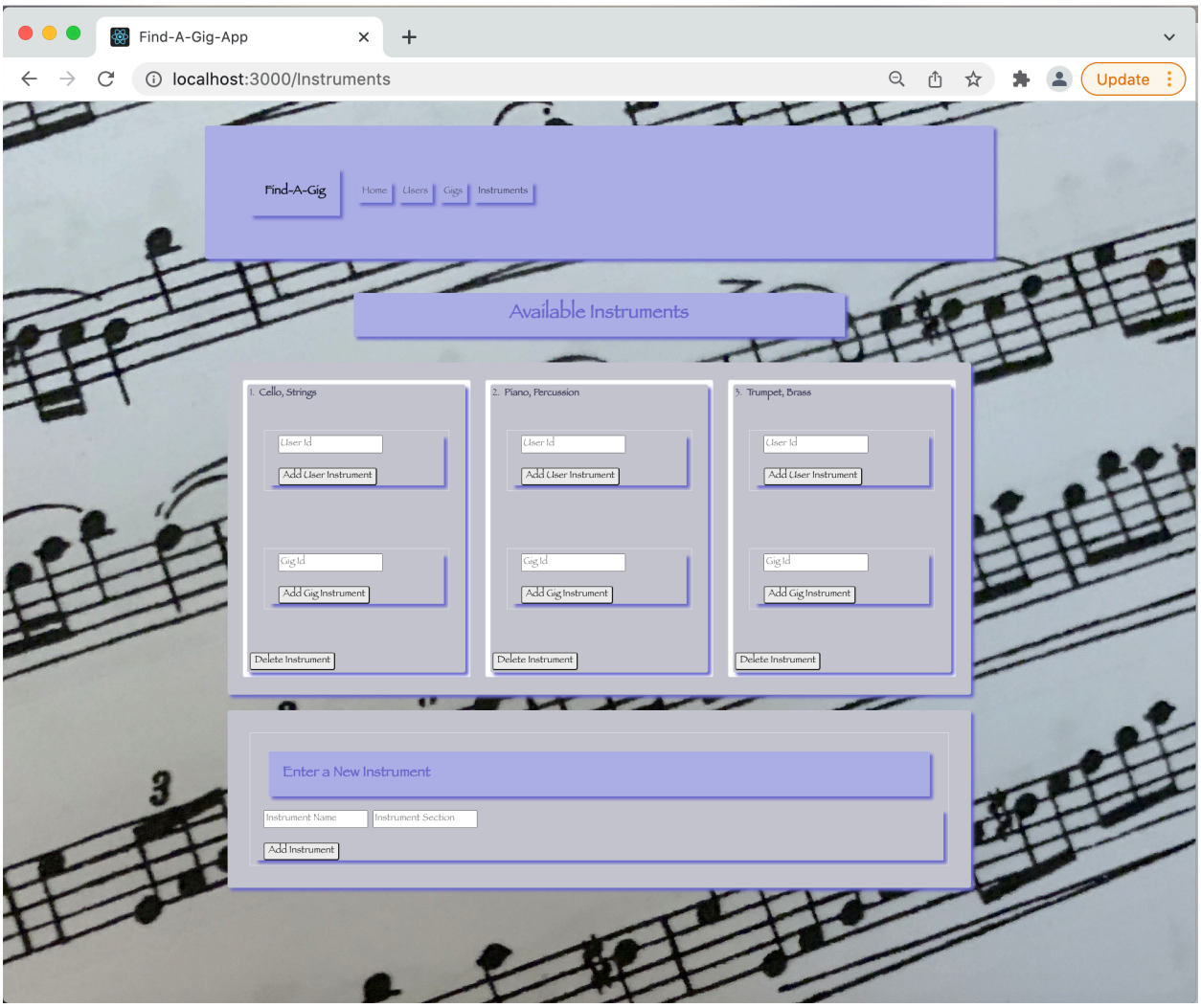


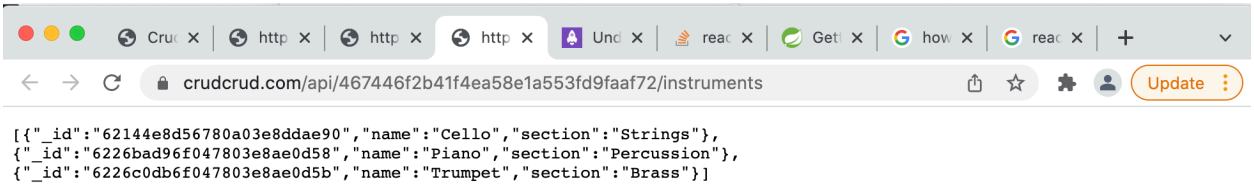

Data stored associated with this screenshot:



PROMINEO TECH

Instruments Page (Read):



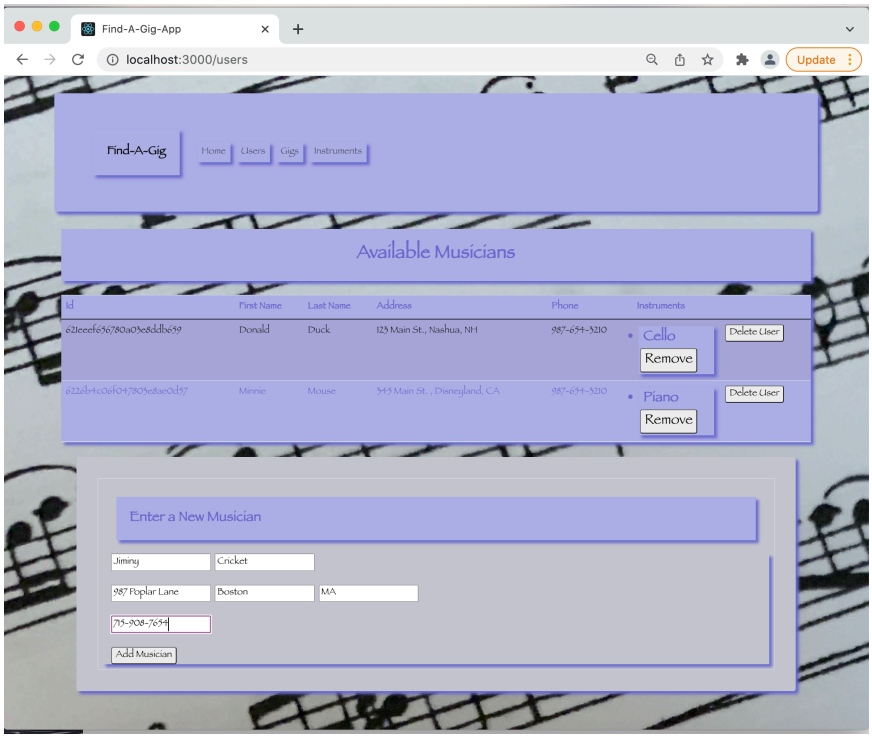





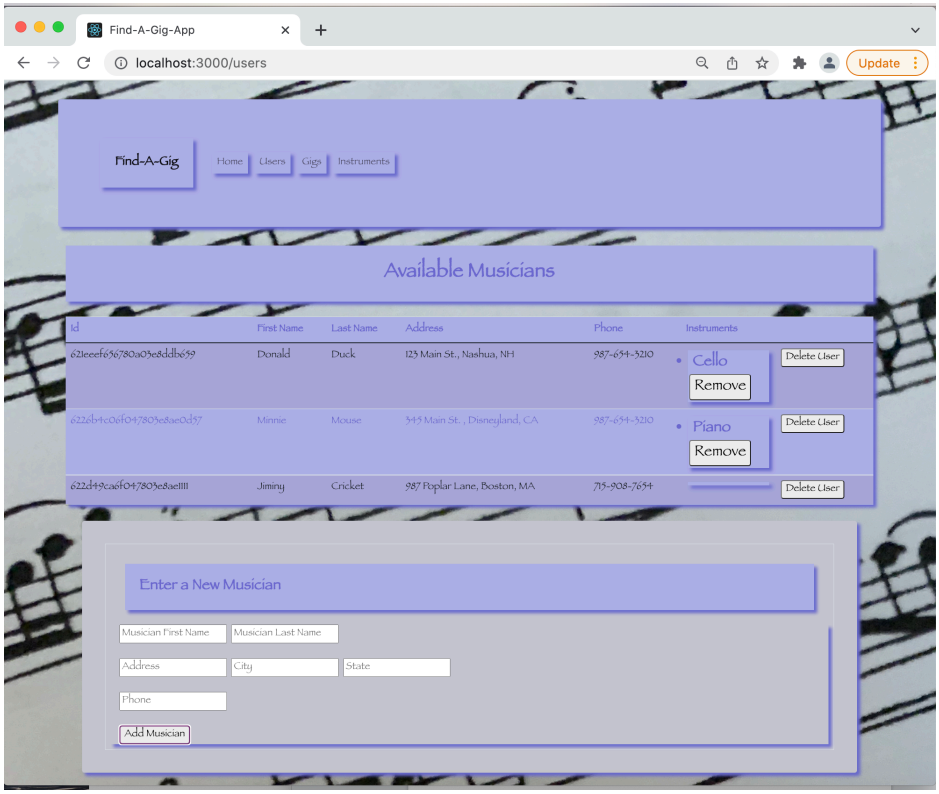
```
[{"_id":"62144e8d56780a03e8ddae90","name":"Cello","section":"Strings"},
{"_id":"6226bad96f047803e8ae0d58","name":"Piano","section":"Percussion"},
{"_id":"6226c0db6f047803e8ae0d5b","name":"Trumpet","section":"Brass"}]
```

PROMINEO TECH

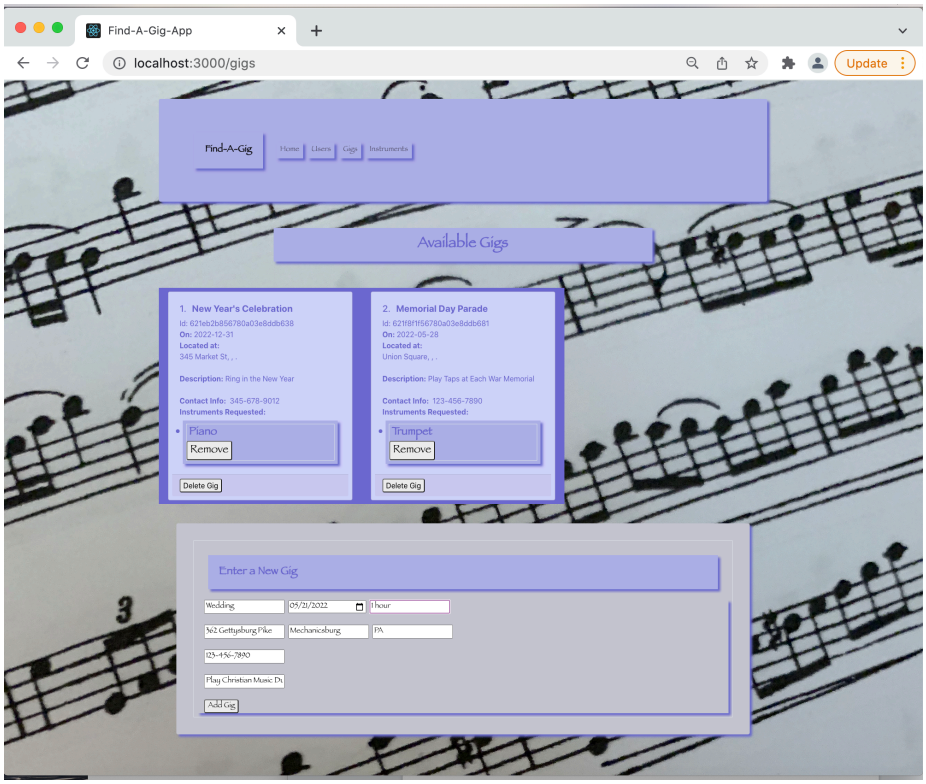
Enter a New Musician (**Create**):



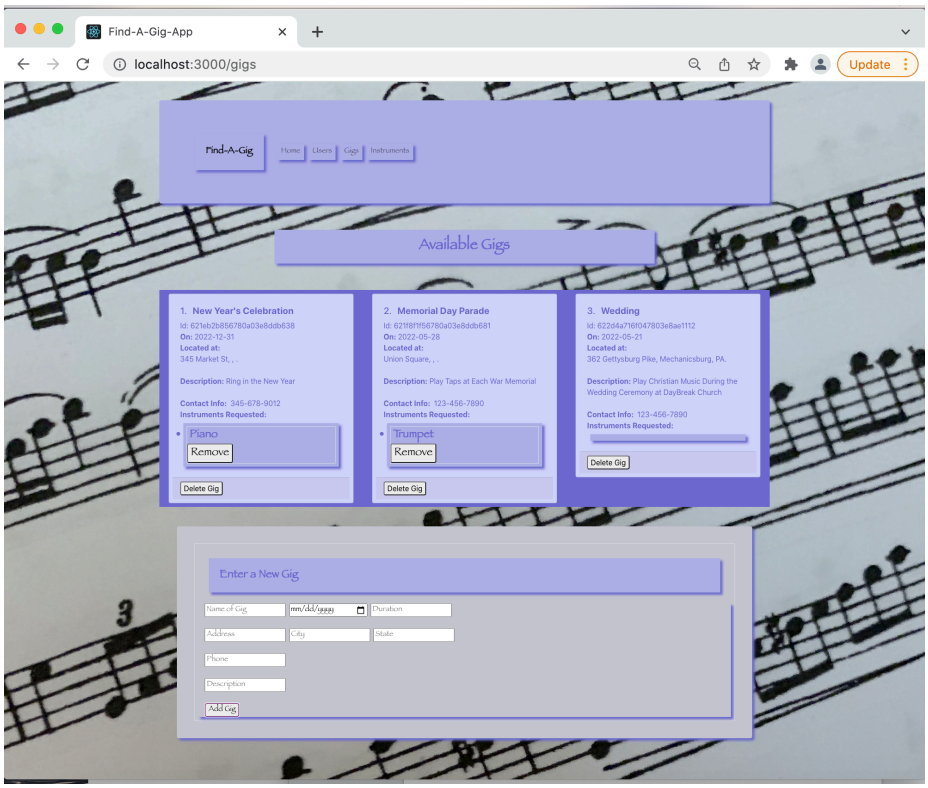

After the **Add Musician** Button is pushed:



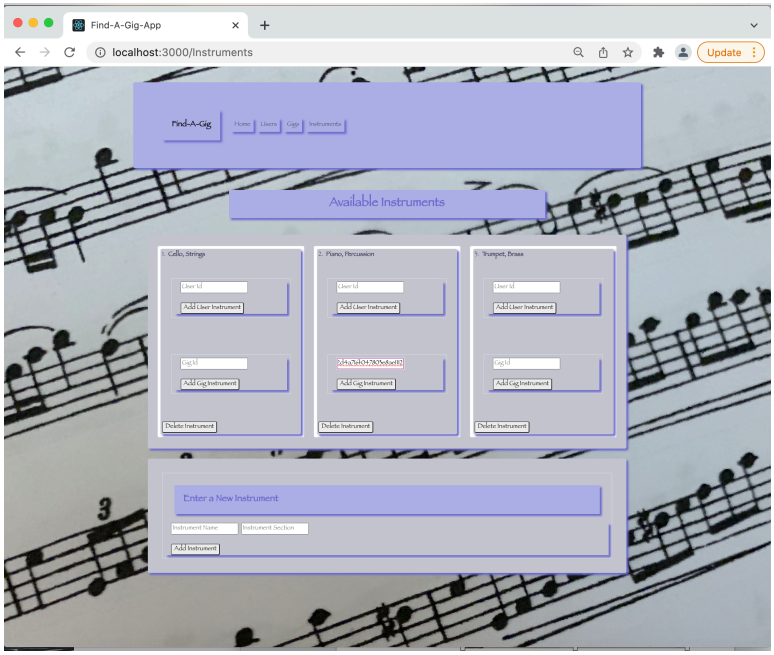
Enter a New Gig (**Create**):

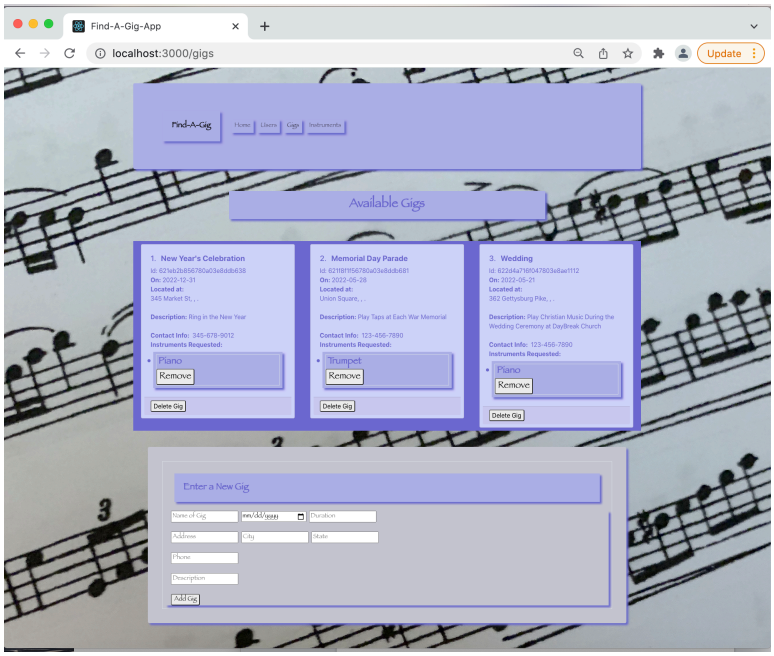


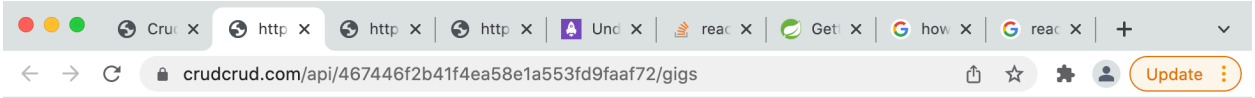

After **Add Gig** button is pushed:



Add an instrument to the Wedding Gig (**Update**):



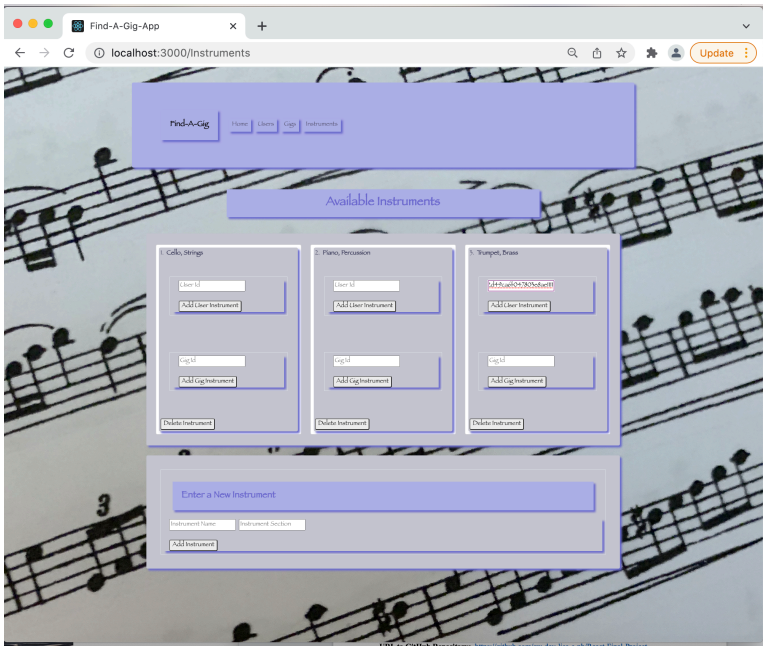



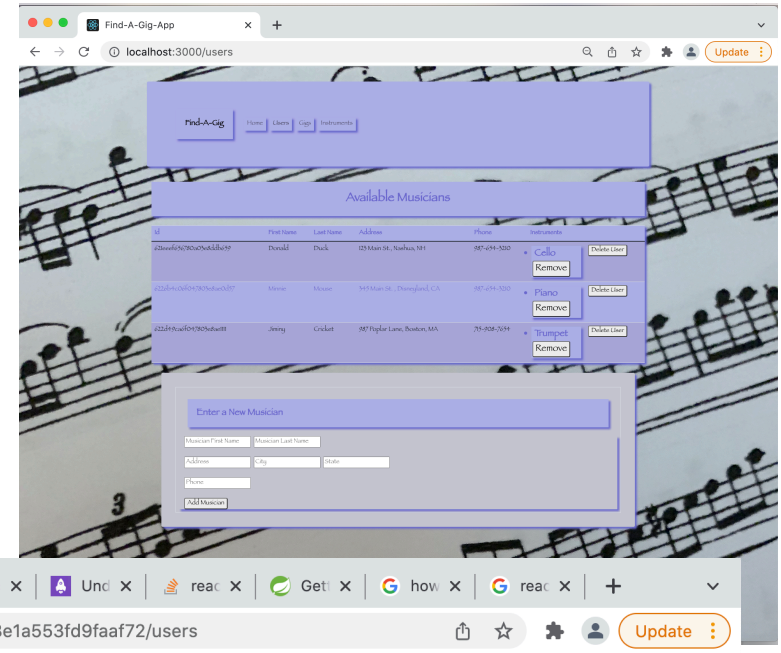





```
[{"_id":"621eb2b856780a03e8ddb638","name":"New Year's Celebration","date":"2022-12-31","address":"345 Market
St","phone":"345-678-9012","description":"Ring in the New Year","instruments":
[{"_id":"6226bad96f047803e8ae0d58","name":"Piano","section":"Percussion"}]},
{"_id":"621f8f1f56780a03e8ddb681","name":"Memorial Day Parade","date":"2022-05-28","address":"Union Square","phone":"123-
456-7890","description":"Play Taps at Each War Memorial","instruments":
[{"_id":"6226c0db6f047803e8ae0d5b","name":"Trumpet","section":"Brass"}]},
{"_id":"622d4a716f047803e8ae1112","name":"Wedding","date":"2022-05-21","address":"362 Gettysburg Pike","phone":"123-456-
7890","description":"Play Christian Music During the Wedding Ceremony at DayBreak Church","instruments":
[{"_id":"6226bad96f047803e8ae0d58","name":"Piano","section":"Percussion"}]}]
```

Add an instrument to User (**Update**):  Jiminy Cricket



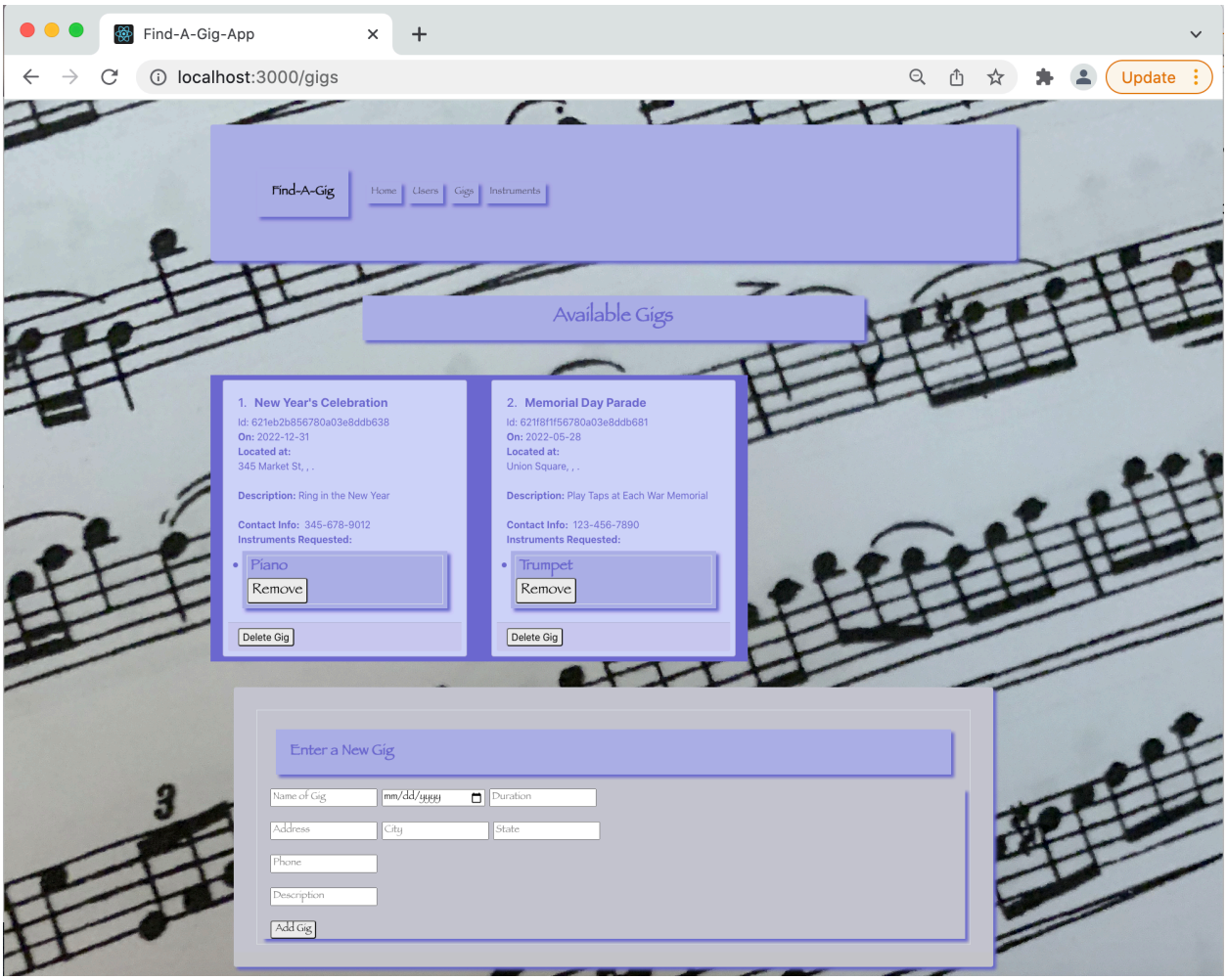



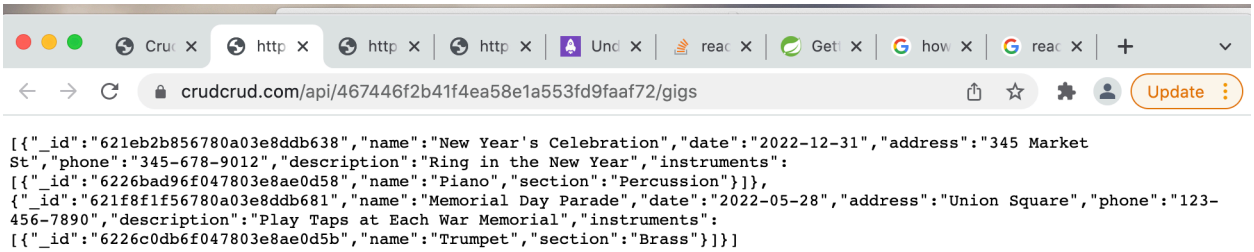





```
[{"_id":"621eeef656780a03e8ddb659","firstname":"Donald","lastname":"Duck","address":"123 Main
St.","city":"Nashua","state":"NH","phone":"987-654-3210","instruments":
[{"_id":"62144e8d56780a03e8ddae90","name":"Cello","section":"Strings"}]},
{"_id":"6226b4c06f047803e8ae0d57","firstname":"Minnie ","lastname":"Mouse","address":"345 Main St.
","city":"Disneyland","state":"CA","phone":"987-654-3210","instruments":
[{"_id":"6226bad96f047803e8ae0d58","name":"Piano","section":"Percussion"}]},
{"_id":"622d49ca6f047803e8ae1111","firstname":"Jiminy","lastname":"Cricket","address":"987 Poplar
Lane","city":"Boston","state":"MA","phone":"715-908-7654","instruments":
[{"_id":"6226c0db6f047803e8ae0d5b","name":"Trumpet","section":"Brass"}]}]
```

Push **Delete** Gig on the Wedding Gig:

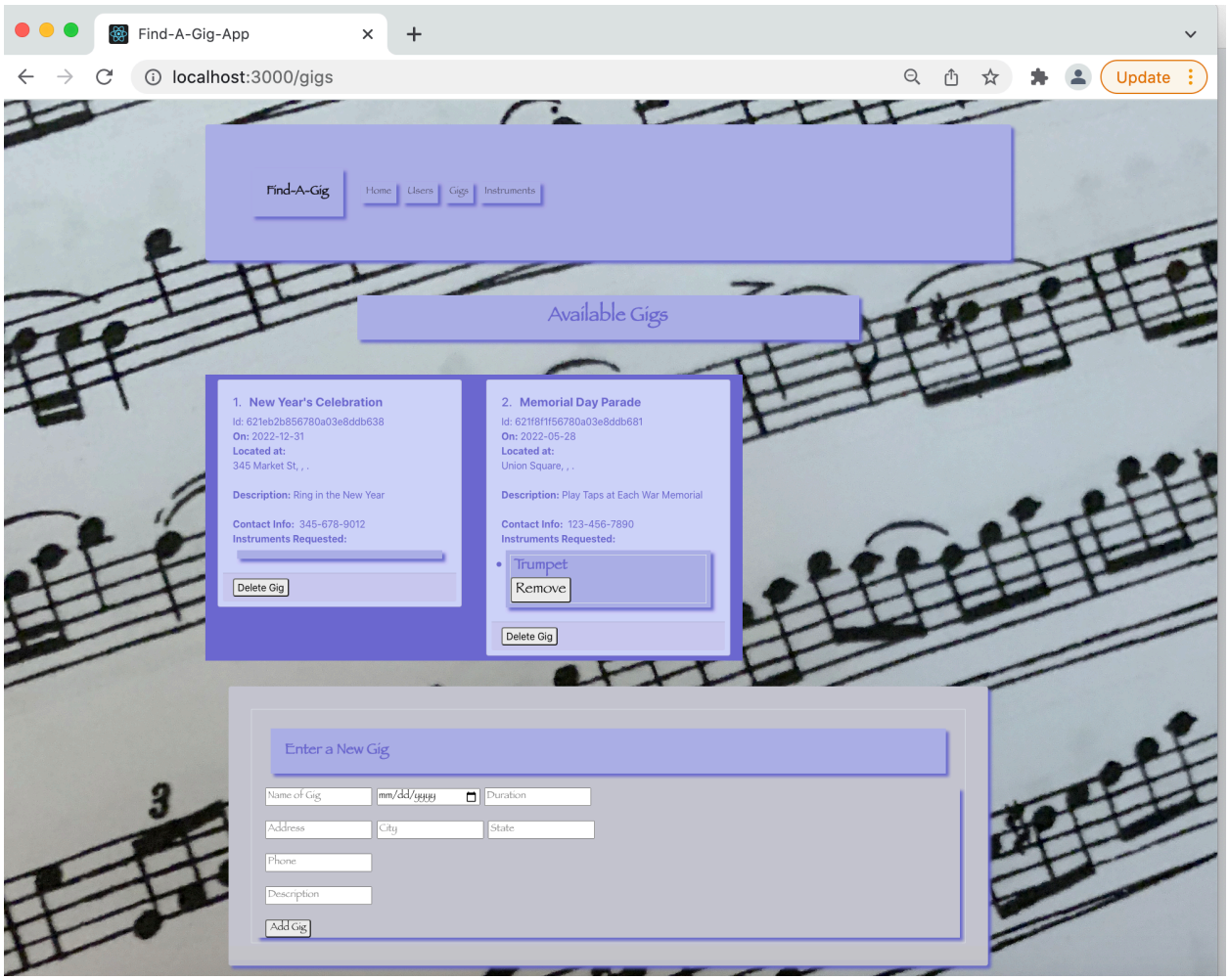


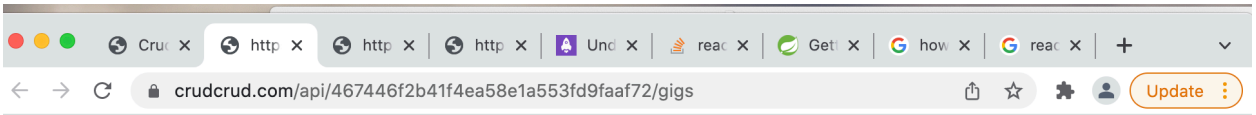





```
[{"_id":"621eb2b856780a03e8ddb638","name":"New Year's Celebration","date":"2022-12-31","address":"345 Market
St","phone":"345-678-9012","description":"Ring in the New Year","instruments":
[{"_id":"6226bad96f047803e8ae0d58","name":"Piano","section":"Percussion"}]},
{"_id":"621f8f1f56780a03e8ddb681","name":"Memorial Day Parade","date":"2022-05-28","address":"Union Square","phone":"123-
456-7890","description":"Play Taps at Each War Memorial","instruments":
[{"_id":"6226c0db6f047803e8ae0d5b","name":"Trumpet","section":"Brass"}]}]
```

Pushed **Delete** Instrument on New Year's Celebration Gig:





```
[{"_id":"621eb2b856780a03e8ddb638","name":"New Year's Celebration","date":"2022-12-31","address":"345 Market
St","phone":"345-678-9012","description":"Ring in the New Year","instruments":[]},
{"_id":"621f8f1f56780a03e8ddb681","name":"Memorial Day Parade","date":"2022-05-28","address":"Union Square","phone":"123-
456-7890","description":"Play Taps at Each War Memorial","instruments":
[{"_id":"6226c0db6f047803e8ae0d5b","name":"Trumpet","section":"Brass"}]}]
```