**Objective:** To have a hands-on experience of working with web applications development using react.js, bootstrap and basics of expressjs.

# Requirements Overview:

Self-Profile Client-side Application with the following requirements:

* React.js, Bootstrap & ExpressJs(preferred) /Spring Boot must be used.
* Practice bootstrap: Should confine to the Specified Template.
* Show proficiency in state variables for user input: As a visitor, I can edit name & description. (Note: persistence of values in a database is optional)
* Show preliminary skills in using 3rd party APIs: As a visitor, I should be able to see the latest news (or any dynamic content!).
* Continue working as an Individual: Regular commits to GitHub’s Classroom repository

are expected.

# Suggested Activities:

**## Understand Web App Structure, JavaScript, Client-Server, REST Communication ## Create a static page and create the required UI**

**## Checkout** [**Creating a Server**](https://github.com/ninadpchaudhari/React-Tutorial#creating-a-server) **Section.**

**## Checkout** [**Exploring React**](https://github.com/ninadpchaudhari/React-Tutorial#exploring-react) **### Understand the theory**

**### Create your first Component**

**### Create your first state-full component.**

**### Add the static HTML page created in this page. ### Ensure the app can be successfully built!**

**## Integrate 3rd party API**

**### Checkout the “news app component” in the “Exploring React”**

**### Understanding concepts about setting state variables (Ref: Checkout way `articles` are being updated.**

**## Deployment**

**### Understand basics of web ports**

**### build and deploy React app using nginx. ### deploy expressjs application.**

**### Make it immortal using pm2.io ## (Optional) Understand AWS S3**

**Requirements:**

1. Import “bootstrap 5.0” and supporting libraries and use the layout systems to create a page replicating the screenshot provided herewith, complete with Navigation bar with Your websites’ name and any logo.
2. Initiate a Spring Boot web project / Expressjs (proffered) that can add 2 numbers.
3. Add your project in git source control system. Push the git repository to GitHub.
4. You are encouraged to do commits in the git repository as frequently as required.
5. Clone your project on your AWS EC2 instance and Host the web application. It should be accessible by anyone with an internet connection. Should work even after you close the “terminal” i.e.,ssh connection
6. The application should communicate with external source to consume a REST End point. news is an example.
7. The react application should have a textbox where the user can input text which will be displayed into specific areas on the page.

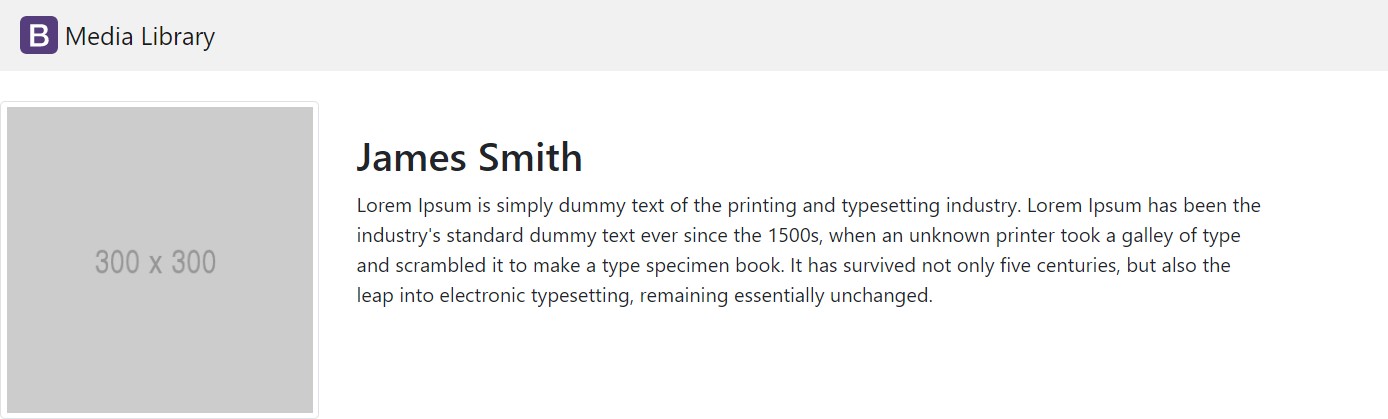
# Instructions to submit your work:

* Create a Single zip file with the name as “YourNetId.zip” with all your work.
* The Folder should contain
  + `One` “readme.pdf” file
  + One Folder file named “source\_code”.
  + One short screen video (under 3-5 mins) demonstrating working and specifying non-working aspects of the application. (A great place to talk about partial progress / issues about a feature that was not implemented)
* Client and Server-side source code should be in separate sub-directories inside the

“source\_code” directory.

* The “readme.pdf” (or readme.md) should have web address of the deployed website (Along with the port, if applicable) and the IP address of the EC2 instance accompanied with a link to github link of your repository.

Annex:



React Tutorial [https://github.com/ninadpchaudhari/React-](https://github.com/ninadpchaudhari/React-Tutorial)

[Tutorial](https://github.com/ninadpchaudhari/React-Tutorial)

Biography

Name