The portfolio application needs to have additional features like getting data from another online application, storing the data in the database, presenting statistics over the data in the frontend, and updating the frontend in real-time when the data changes, you can follow these steps:

1. Set up your development environment by installing NodeJS and MongoDB on your local machine.

2. Create a new NodeJS project by running npm init in your project directory. This will create a new package.json file, which will be used to manage your project dependencies.

3. Install the necessary dependencies for your project. You will need to install the following packages:

* express - A NodeJS web framework that will be used to handle HTTP requests and responses.
* socket.io - A real-time communication library that will be used to implement the WebSocket server and real-time messaging.
* mongoose - A MongoDB driver for NodeJS that will be used to interact with your database.
* request - A library for making HTTP requests to other online applications.
* react and react-dom - The core libraries for building React applications.

You can install these packages using the npm install command.

4. Set up your server-side code by creating an Express.js application. This will include setting up routes, middleware, and error handling.

5. Use the request library to get data from another online application. You will need to make HTTP requests to the API endpoint of the online application and parse the response data.

6. Store the data in your MongoDB database using Mongoose. You can create a schema for your data and use Mongoose to insert or update documents in the database.

7. Implement a WebSocket server using Socket.io. This will enable real-time communication between the client and the server.

8. Implement a real-time statistics feature using Socket.io. You can create a dashboard or visualization that presents statistics over the data in the database, and use Socket.io to update the dashboard in real-time when the data changes.

9. Set up your client-side code using React. This will include creating React components, handling events, and rendering the user interface.

10. Implement a form or interface that displays the real-time statistics generated in step 8.

11.Deploy your application to a cloud hosting platform like Heroku or AWS Elastic Beanstalk to make it accessible over the internet.

12. Include comments at the beginning of each code file to describe what it does.

By following these steps, you should be able to build a portfolio application that gets data from another online application, stores the data in the database, presents statistics over the data in the frontend, and updates the frontend in real-time when the data changes using NodeJS, React, and MongoDB.