APPLICATION #6: **REAL-TIME MESSAGING APP (CAPSTONE PROJECT #2)**

**Description**

A chat application. Think of something like Slack.

People will need to register for an account and will need to log in to use the app. The chat history is stored and retrieved from a database. You can add notifications (pop up or sound) in the browser for when someone sends a new message. If you are looking for an extra challenge, you can create a simple chat-bot that responds to certain commands.

Example: <https://naamp-chat-app.herokuapp.com/>

**Due Date**

3 weeks

**Requirements**

* Uses React
* Uses Bootstrap for styling/responsiveness
* Uses C# (.NET Core)
* Uses NHibernate as an ORM (Object Relational Mapper)
* Uses PostgreSQL
* Uses WebAPI for Internal API
* In-memory Unit tests are written
* Deploy using Microsoft Azure

**Deliverables**

* Basic app flow:
  + When opening to the root of the site, you will be directed to the login screen if not logged in.
    - Login screen should have an option to register a new user.
  + A user will be able to send messages
  + A user can read all messages in the main chat room
* Functionality:
  + All chat messages will be saved in a SQL table.
  + When sending a new message in the chat, a new chat record will need to be created in the database.
* Database structure:
  + User table
  + Messages table
  + Channel table
* Unit tests should cover:
  + Creating a new chat message
  + Creating a new user
  + Modifying user data

**Learning Resources**

* **Technology/Programming Languages**
  + **Week 1**
    - Chat App - High Level Overview: [Link](https://www.dropbox.com/s/qdz2rd6yly754zw/High%20Level%20Overview.mp4?dl=0)
    - Week 1 Outline Video: [Link](https://www.dropbox.com/s/ki98jfcoxvk7190/Chat%20App%20Week%201.mp4?dl=0)
  + **Week 2**
    - Week 2 Outline Video: [Link](https://www.dropbox.com/s/g7g3qsyqnh5tag2/Chat%20App%20Week%202.mp4?dl=0)
  + **Week 3**
    - Week 3 Outline Video: [Link](https://www.dropbox.com/s/znu1iwxeu5bm1wg/Chat%20App%20Week%203.mp4?dl=0)
* **Technology/Programming Languages**
  + React
    - Learn React Course (Scrimba): [Link](https://scrimba.com/g/glearnreact)
    - React “Getting Started” Documentation (Very helpful): [Link](https://reactjs.org/docs/getting-started.html)
  + C#
    - Tour of C# (MSDN Documentation): [Link](https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/index)
    - Head First C# Book (Optional): [Link](https://www.amazon.com/Head-First-Learners-Real-World-Programming/dp/1449343503/)
  + SQL
    - Very good high level overview of SQL: [Link](https://learntocodewith.me/posts/sql-guide/)
    - Head First SQL: Your Brain on SQL Book (Optional): [Link](https://www.amazon.com/Head-First-SQL-Brain-Learners-ebook/dp/B009VQ7NIQ)
    - Khan Academy course on SQL: [Link](https://www.khanacademy.org/computing/computer-programming/sql)
  + NHibernate
    - What is an ORM (Object Relational Mapper)? [Link](https://stackoverflow.com/questions/1279613/what-is-an-orm-and-where-can-i-learn-more-about-it#answer-1279678)
    - NHibernate QuickStart: [Link](https://www.tutorialspoint.com/nhibernate/nhibernate_quick_guide.htm)
    - NHibernate Intro and Configuration Tutorial: [Link](https://www.codeproject.com/Articles/1178251/NHibernate-Introduction-configuration-CRUD-with-MV)
  + PostgreSQL
    - Installing and Creating First Database (Video): [Link](https://www.youtube.com/watch?v=xaWlS9HtWYw)
  + Authentication
    - What is a JSON Web token in 5 steps: [Link](https://medium.com/vandium-software/5-easy-steps-to-understanding-json-web-tokens-jwt-1164c0adfcec)
    - JSON Web tokens introduction: [Link](https://jwt.io/introduction/)
    - Setting up Authentication in React w/JSON web tokens: [Link](https://medium.com/@romanchvalbo/how-i-set-up-react-and-node-with-json-web-token-for-authentication-259ec1a90352)
  + .NET Web API
    - Get Started with ASP.NET Web API 2: [Link](https://docs.microsoft.com/en-us/aspnet/web-api/overview/getting-started-with-aspnet-web-api/tutorial-your-first-web-api)
    - Introduction to the ASP.NET Web API: [Link](https://www.pluralsight.com/courses/aspnetwebapi)
  + Websockets
    - Websockets explanation and .NET Core: [Link](https://www.c-sharpcorner.com/article/how-to-use-websockets-in-asp-net-core-day-nine2/)
  + NUnit Library
    - Documentation Page: [Link](https://github.com/nunit/docs/wiki)
    - Introduction to .NET Testing with NUnit (Pluralsight): [Link](https://www.pluralsight.com/courses/nunit-dotnet-testing-introduction)
    - Thorough explanation of unit testing with NUnit: [Link](https://www.codementor.io/copperstarconsulting/intro-to-unit-testing-c-code-with-nunit-and-moq-part-1-y2b9iv8iq)
  + Bootstrap
    - Learn Bootstrap 4 (Scrimba Tutorial): [Link](https://scrimba.com/g/gbootstrap4)
  + Deploying to Azure
    - Set up a free account and get 12 months of freebies: [Link](https://azure.microsoft.com/en-us/free/free-account-faq/)
    - Getting Started with Azure Cloud Services and ASP.NET: [Link](https://docs.microsoft.com/en-us/azure/cloud-services/cloud-services-dotnet-get-started)

Creating a Postgres database in Azure: [Link](https://docs.microsoft.com/en-us/azure/postgresql/quickstart-create-server-database-portal)