Modules Implemented

Client (Frontend):

- Navigation Bar - different for mobile and desktop.
- Login/Signup Page - Authentication and Authorization
- Contact Form
- Coverflow Slider for News Display
- Story Mode, Situation, Explanation and Supporting Modules
- Chatbot with inbuilt Sentiment Analysis - borrowed and restructured code from here.
- User Profile - supporting modules to handle images and dark/light modes.

Server (Backend):

- Router Module - Used Express.js to create a complete lightweight web framework with a waterfall model of below modules
- User Module - to handle registration, editing and deletion of user profiles
- Authentication Module - to issue, validate and update user session keys
- Image Handling Module - to save to Cloudinary and send the url to be saved in MongoDB
- News Module - to respond to news requests from the client, while using an external, replaceable API source
- Story Module - to handle requests from client and source all information and state, regarding stories for a user
- Translation Module - to handle all content translations with an external, replaceable API
- MongoDB & Mongoose Helper Module - to create an object-relational mapping (ORM) for MongoDB to store the application data
- JWT Module - to validate the user's JSON Web Token, which improves security

Modules Not Implemented

There were no roadblocks but the features below could not be completed

- Community Chat/Discussion & related helper modules: The user participates in a community chat/discussion platform to share their experiences and learn from others. Could not complete because of time constraints.

- Sentiment Analysis: The application does not currently have a full-fledged Sentiment Analysis feature. The current one is a bit primitive. This was because we could not find such an open-source implementation which does not diverge from the context.