# Introduction

In a world where disasters are now a common occurrence we still have a lot to do in emergency preparedness and management front. We are very advanced but there is still room for improvement. This hackathon is an effort to add to the current systems for emergency preparedness.

# Problem

For emergency preparedness, there is no robust system to host the people data in a centralized manner. It is unclear if emergency workers can identify with an accuracy how many people could be affected by a certain emergency and how many of them are children, need medical assistance or need medical transport. During an emergency if a family is displaced, it is very hard to locate the family members. Volunteers with special skills belonging to a CERT may be needed more at a certain shelter, but there is no way to identify that need.

# Solution (Initial attempt)

Our EPMS solution is a small step towards building a system that addresses the above problem areas. Using our solution an individual can register their family members and find shelter in the event of an emergency. They can even search for a shelter which has specific amenities, such as, child friendly, elder care and special medical needs. A volunteer with specialized skills can identify the shelter where there is a shortage of that skill. Using Family Map, displaced family members can find their family members. This is done when individuals are checked in to a certain shelter. The data captured during preparedness registration can be used to assess the amount of personnel and supplies needed.

Our solution could be accessed at <http://rei-velocity.azurewebsites.net/>

# Development Technologies

EPMS is developed using MongoDB, Express JS and Node JS. Jade was used as the rendering engine. The website is hosted on Azure as a webapp. We have created an Ubuntu VM on Azure which holds the data in MongoDB. The data is exposed in RESTful API using Mongoose ORM. This EPMS webapp uses the web APIs hosted in the Ubuntu VM. The webapp uses twitter bootstrap to achieve responsive design. We have used Sockets IO for real-time communication with the clients.

# Current Limitations

Due to the limited amount of time we were not able to load the data from the dataset to assist in searches. Currently this is being submitted a very robust framework which provides a platform to build a complex system in very less time.