This document is intended to outline the initial user stories, website template, front-end and back-end models, and basic database concepts.

***User Story: Login/Sign-Up***

*“As a user, I would like to sign-up to this website, and login when desired with my email and password”*

    The user requires a login and sign-up system. For the initial phase, this login and sign-up will be very basic in order to follow the paradigm of incremental development. The login/sign-up forms will be on the same page, where the left-hand side includes the login system of email and password, and the right-hand side will include the sign-up system of email and password. The databases will store these values, along with other user properties that will be added to the sign-up in a future user story.

During sign-up and login, the email should be validated via simple *html based validation*, no JS at this point as it is unnecessary.

In the sign-up system, once the email is validated, the password should be validated via simple character limit requirements. If both email and password are valid, the email is checked against the user-login database to see if the email already exists. If it doesn’t, then a new account is added to the database, and the user is notified of this with an appropriate message. If the email is already taken, then the user is notified of this and nothing occurs.

For the login system, the email is checked for validity, and if it is valid, the email is used to query the database. If the email exists in the database, it’s password is checked against one provided by the user. If the email and password provided by the user match an equivalent pair in the database, the user is logged in and notified of this; else the user is notified of the incorrectly provided data.

***User Story: Post a Question***

*“As a user, I want to be able to post a question once logged in to my own personal account”*

The question page should have a *title* text box with a 100-character limit, and a question *body* with a 1000-character limit. No submissions will go through unless both boxes contain content. These requirements will be refined as the project moves ahead.

If the user is not logged in, the page should display the option for the user to login/sign-up before posting. If the user is logged in, he/she will be able to send a question. Once a valid question is submitted by a logged-in user, the title and body will be sent to the server side via a *post*. Along with the title and body of the question, several other auto-generated metrics (date,time,user\_id) will be appended to the entry and stored in an appropriate database. A message is returned telling the user the question was successfully submitted.

***User Story: Answer a Question***

*“As a user, once I find an appropriate question through a search (in previous user story), I want to be able to answer the question”*

Each question will have its own dedicated page that lists the question title and body, and all previous answers to that question below it, and at the bottom of it all, a text area of a logged-in user to submit his/her own answer. The answer should be a maximum of 1000-characters, and an empty answer will not be submitted. If the user is not logged on, there should be an appropriate redirect button to allow the user to quickly login/signup otherwise they will not be able to login.

Once a valid answer is submitted, the answer will be stored in an appropriate server-side database with a few auto-generated fields (date, time, user\_id, score…). When the data is finally stored on the server, the server is return the same page updated with the newly submitted answer.