Assumptions:

- ***Recheck Past Searches*** - functionality assumes user wants to recheck prior results for any additional recalls, rather than recall earlier search criteria to run identical or similar new search.
- ***Sequential user access from single device*** - session state is implemented in such a way that user login from a new device expires all prior logins from other devices.
- ***Limited results*** - current FDA API specifications return a maximum of 100 results in a single query. This limit is assumed to be sufficient for the purposes of Dangerous Dinner app.
- ***Current login is limited to username and password combination*** - it is assumed no additional personal information needs to be stored in the system.
- ***Not optimized for Mobile Devices*** - current version is mobile device compatible for usability and is fully functional, however certain layouts needs to be adjusted to make it production grade.

Further Enhancements:

- ***Implement proper logging*** - add performance logging for statistics gathering to identify potential bottlenecks and areas for improvement. Add access logging for security and to be able to trace back system activities in case of crashes or breaches.
- ***Convert to HTTPS*** - currently built on HTTP, Dangerous Dinner app is susceptible to eavesdropping/man-in-the-middle attacks. While the password itself is encrypted, an attacker can still gain access to prior search results for victim user.
- ***Unit Testing*** - currently not implemented, but this would be a certain “must” to enhance quality and augment build-release cycle.
- ***UI: enhance results rendering*** - currently results are being returned directly from API and rendered in paragraph fashion. To make it much more usable for end client, results could be broken up by attributes and rendered in a fashion that would allow filtering and sorting (for example by state/zip). This would allow to greatly enhance result relevancy for the user.
- ***Bug: intermittent excess scrolling*** - certain scenarios result in a vertical scrollbar on pages that do not require a scroll bar at all. Low severity issue, as it only happens in specific cases and only impacts visualization (not functionality). Nonetheless, would make sense to address to make production grade.

Backend Server:

Windows 2012 R2 Datacenter VM running in Google Public Cloud (GPC). VM has MySQL Server installed, that hosts the Dangerous Dinner database. The VM also has Node.js installed running ***routes.js*** on port 3000, which listens and accepts web calls and transforms them to MySQL parameterized calls. Built React app is then hosted through IIS, running on port 80.