1. [Node.js](https://nodejs.org/en/" \t "_blank) - Server-side Development
2. [Express.js](https://expressjs.com/) - REST API creation for different Routes.
3. [Handlebars.js](http://handlebarsjs.com/) - Frontend templating
4. [MongoDB](https://www.mongodb.com/)- NoSQL database to store chat and user meta info
5. [Firebase Cloud Messaging](https://firebase.google.com/docs/cloud-messaging/) - Push Notification
6. [Socket.io](http://socket.io/) - Real-time data transferring from client-server and vice-versa
7. [OpenPGP.js](https://github.com/openpgpjs/openpgpjs) - End-to-End Encryption for messages
8. [Passport.js](http://passportjs.org/) - Authentication Middleware for login
9. [NodeMailer](https://nodemailer.com/) - Send e-mails with Node.JS
10. [Mailgun](http://www.mailgun.com/) - Use with Nodemailer as an email transport middleware.
11. [Nginx](https://www.nginx.com/) - Use for TCP and UDP Load Balancing & Proxy etc.

Reference:

1. <https://www.quora.com/What-kind-of-technology-does-Slack-use>
2. <https://www.markupbox.com/blog/angularjs-vs-reactjs-vs-nodejs-which-one-is-best/>
3. <https://medium.com/@thinkwik/web-sockets-vs-xmpp-which-is-better-for-chat-application-113e3520b327>
4. <https://db-engines.com/en/system/Firebase+Realtime+Database%3BMongoDB%3BRethinkDB>
5. <https://db-engines.com/en/system/Cassandra%3BFirebase+Realtime+Database>
6. <https://www.crazyengineers.com/threads/which-technology-stack-would-you-recommend-for-real-time-chat-system.88054/>
7. <https://www.applozic.com/blog/which-technologies-should-be-considered-for-making-chat-applications-like-messenger-and-whatsapp/>