

NAREN HEMNANI

SOFTWARE DEVELOPER



PERSONAL PROFILE

I am a software developer with robust problem-solving skills and proven experience in creating and designing software in a test-driven environment.

ACADEMIC HISTORY

School of Accelerated Learning Product Engineering Track

Completed and deployed 11 MVPs using technologies related to JavaScript.

Guru Nanak Institutions of Technical Campus

B.Tech Computer Science

CONTACT INFO

Email: naren.khmn@gmail.com

Phone: 7675966900

TECH COMPETENCIES

JavaScript	●	●	●	●	●	●	●
HTML/CSS	●	●	●	●	●	●	●
Bootstrap	●	●	●	●	●	●	●
Node JS	●	●	●	●	●	●	●
Express JS	●	●	●	●	●	●	●
MongoDB	●	●	●	●	●	●	●
Postgresql	●	●	●	●	●	●	●

- Mongoose
- Sequelize
- JWT
- Express Sessions
- Web-sockets
- MVC Architecture
- OAuth
- AWS

PROJECTS

SHARING BENEFITS (CAPSTONE PROJECT)

Node.JS, Express.JS, Express-Handlebars, OAuth, MongoDB

This App allows users to share ongoing shopping offers. A user can create a post with offer details and find a companion who is interested in the offer. Both users can shop for the min threshold bill value to avail the offer and can share the benefits. This App also displays the current offers running in the market.

IMDB CLONE

Node.JS, Express.JS, Express-Handlebars, JWT, Postgresql

Developed a clone of the popular movie database with an option for user to add a movie and its cast. Users can browse movie ratings, cast, and view cast details.

HACKER NEWS CLONE

Node.JS, Express.JS, EJS, Passport-local, MongoDB

Simulated a clone of the Hacker News website where users can view Tech related posts, Login or Register and can also submit a post. User can edit or delete the post he submitted.

E COMMERCE APPLICATION

Node.JS, Express.JS, EJS, JWT, Bootstrap, Postgresql

Built an E-commerce application where users can purchase products. Incorporated user Authentication, view Products by Category, Add to Cart and Place Order options.

CHAT APPLICATION

Node.JS, Express.JS, Socket.io, JWT, Postgresql

Created a Chat Application which allows users to Login or Register and Chat with other Users. The sent and received messages are simple Post and Get requests, here Socket.io allows us to perform Get and Post requests without refreshing the route.