Jaipur Engineers
Your One-Stop IT & Staffing Ecosystem

FULL STACK

MERN Stack / Full Stack with Node.js

MongoDB • Express.js • React.js • Node.js



COMPLETE COURSE

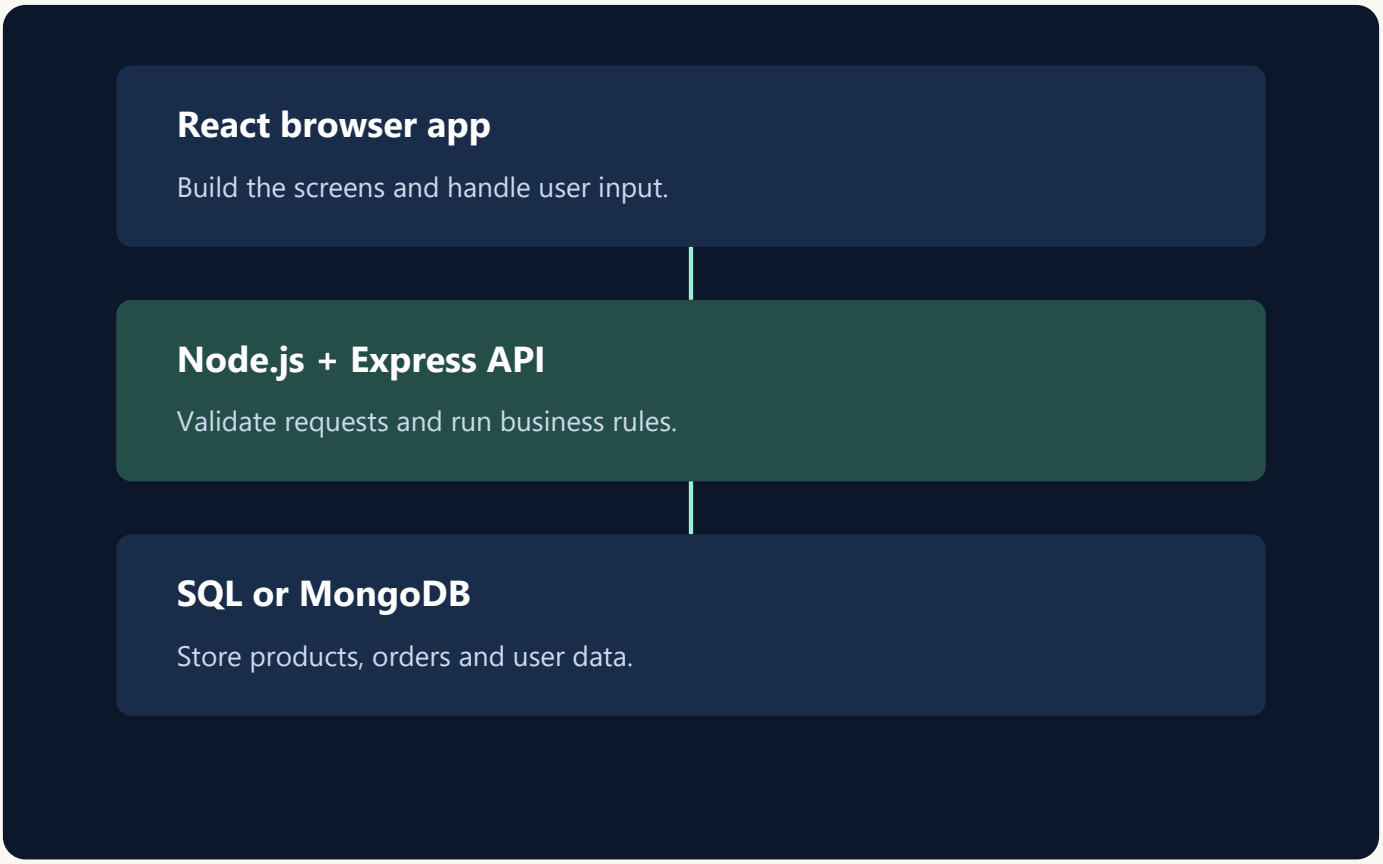
₹28,500

JAIPUR, RAJASTHAN

+91 70146 92039

How the stack fits together

Follow a product search request from the browser to the database and back.



Frontend

Build responsive pages and interactive screens.

Backend

Create API routes and protect application data.

Delivery

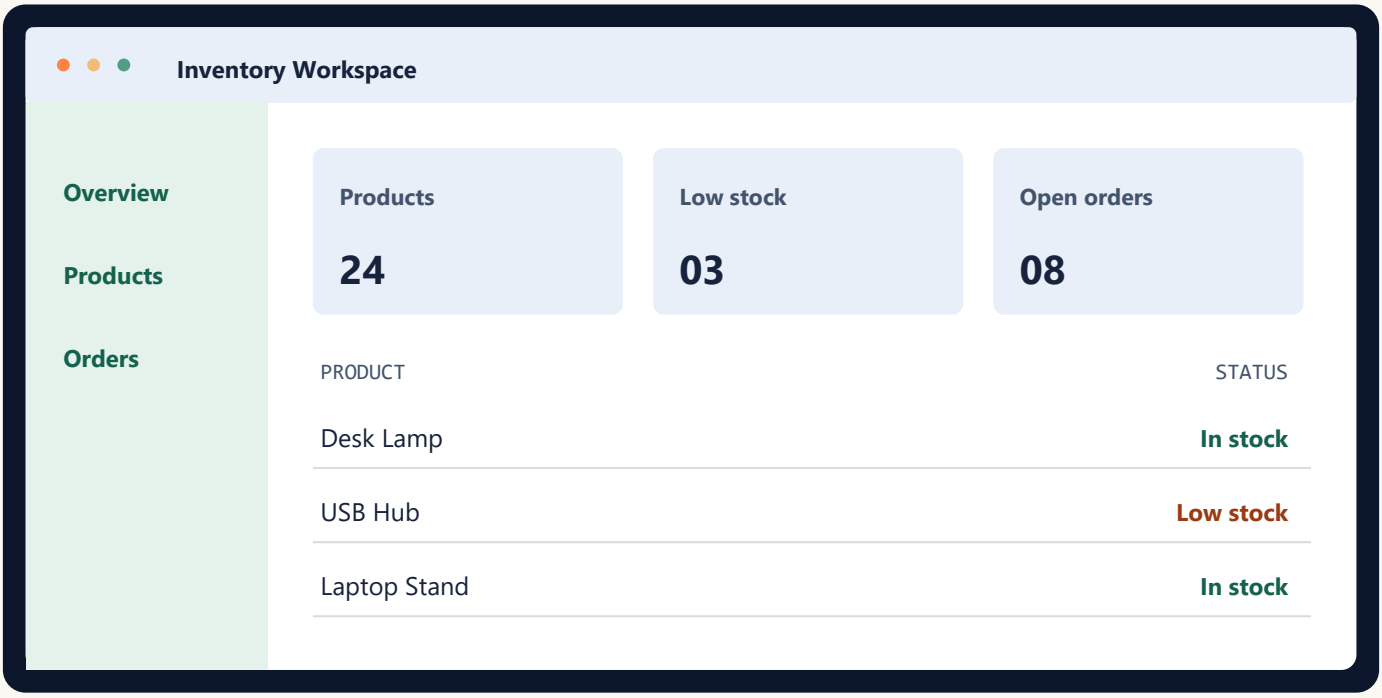
Connect the layers, test the workflow and prepare a deployment.

WHO IT'S FOR

Suitable for students and developers moving into full-stack web development. Familiarity with basic coding helps.

Build responsive web pages

Start with semantic HTML, flexible CSS and forms users can navigate with a keyboard.



ILLUSTRATIVE INVENTORY DASHBOARD

HTML5

Structure pages with landmarks, tables, links, inputs and form labels.

CSS3

Use the box model, Flexbox, Grid and breakpoints for responsive layouts.

CSS frameworks

Compare Bootstrap components with Tailwind CSS utility classes.

PRACTICE

Build a product list that works on desktop and narrow screens. Inspect layout and focus states in browser DevTools.

Make pages interactive

Use JavaScript to validate forms, transform records and bring API data into the interface.

products.js

```
async function loadProducts() {
  const response = await fetch('/api/products');
  if (!response.ok) throw new Error('Could not load products');
  return response.json();
}
```

Core JavaScript

Variables, arrays, objects, functions, DOM events and form handling.

Modern syntax

ES6 to ES10 features, modules, destructuring and array methods.

Async workflow

Promises, async/await, JSON and handling unsuccessful requests.

PRACTICE

Fetch product records and show separate loading, empty and error states.

Build a React application

Turn the dashboard into reusable components, routes and predictable state.

01

Components, props and hooks

Build reusable tables and forms with JSX. Manage component data with useState and coordinate side effects with useEffect.

02

Routes and screen states

Use React Router for product, order and detail screens. Handle form feedback and missing records.

03

Redux and Redux Toolkit

Organise shared state with slices, selectors and async request status. Keep small UI concerns local to the screen.

PRACTICE

Filter inventory by stock status and show the selected product on a detail route.

Build the server

Turn incoming requests into validated actions and return consistent JSON responses.

products-route.js

```
app.get('/api/products', async (req, res, next) => {
  try {
    res.json(await products.list());
  } catch (error) { next(error); }
});
```

Node.js runtime

Explore npm, modules, the event loop and asynchronous I/O.

Express routing

Organise routes, middleware and request handlers by feature.

Configuration

Manage environment settings and keep secrets outside the repository.

PRACTICE

Add a products route and return a clear response when a request cannot be completed.

Define the API contract

Design resource URLs and response formats that the React app can depend on.

METHOD	RESOURCE	PURPOSE
GET	/api/products	List and filter products
POST	/api/products	Create a product
POST	/api/orders	Place an order
PATCH	/api/orders/:id/status	Update an order status
DELETE	/api/products/:id	Remove a product

Validate at the boundary

Validate request data, choose HTTP status codes, and use Postman to check both successful and invalid requests.

PRACTICE

Send a valid order, then submit one without products and confirm the API returns a useful validation error.

Model relational data

Connect products and orders through order items. Use constraints to keep records consistent.

Products

id · primary key

name

price

stock_qty

Order items

order_id · reference

product_id · reference

quantity

unit_price

Orders

id · primary key

customer_id

status

created_at

Each order item links one order to one product.

Querying

Write SELECT, JOIN, GROUP BY and filters for common reports.

Integrity

Work with keys, constraints, transactions and indexes.

Engines

Compare MySQL, PostgreSQL and Oracle Database at a conceptual level.

PRACTICE

Use a JOIN to show each order with its products and calculate the order total.

Compare data stores

Practise document-based CRUD with MongoDB and see where other database models fit.

MongoDB

Work with collections and documents. Model products, filter records and review indexes.

Cassandra

Compare a wide-column model. See how partition keys affect query design.

DynamoDB

Explore a managed key-value and document database and how it uses access patterns.

Apache Druid

Explore a store designed for fast analytics over event and time-series data.

PRACTICE

Store and filter product documents in MongoDB. Compare the structure with the relational model.

Protect the workflow

Use the API to decide who can see an order or change inventory levels.

01

Sign-in and sessions

Understand password hashing and session or token-based authentication. Handle sign-out and expired sessions.

02

Permissions and validation

Check permissions in server-side handlers. Use parameterised queries and validate data before storage.

03

Connect React, API and database

Load data through the API, send form changes and handle validation or permission errors in the browser.

PRACTICE

Verify that staff can update inventory while a signed-out visitor cannot submit changes.

Prepare the app for release

Check the full workflow before deploying the frontend, server and data connection.

01

Track changes with Git

Use branches, clear commits and a GitHub repository. Document local setup and environment variables.

02

Test across layers

Check responsive views, API response codes and database changes. Add tests around important application rules.

03

Deploy and diagnose

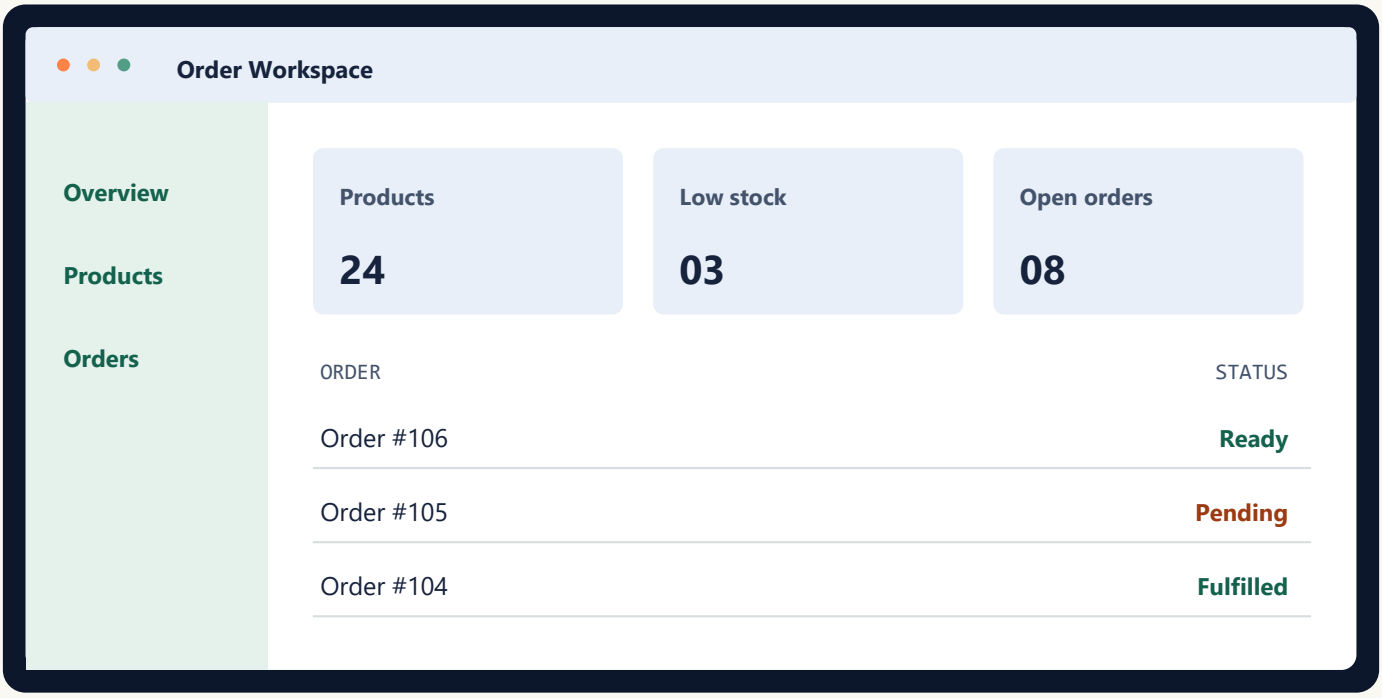
Configure HTTPS, application environment and database connectivity. Check logs and run a short smoke test after deployment.

PRACTICE

Deploy a preview, create a test order and confirm the dashboard and database reflect it.

Inventory and order portal

Build an internal web app that tracks products, accepts orders and shows what needs restocking.



ILLUSTRATIVE ORDER DASHBOARD

Product catalogue

Search products, display available stock and edit records from the dashboard.

Order workflow

Create an order, validate stock and update status through the API.

Reporting

Show low-stock products and open orders using database queries.

PROJECT DEMO

Create an order, show the stock change and trace how the update reaches the dashboard.

Present your project

Use the application to explain what you built, how it works and where you made design choices.

01

Make it reproducible

Publish the code with a README, setup steps, sample data and the environment variable names the app expects.

02

Record a short walkthrough

Show a working product list, an order submission and how the UI responds when a required field is missing.

03

Explain the decisions

Trace a browser request through Node.js to the database and explain why you chose a relational or document model.

INTERVIEW PRACTICE

Use examples from your project when preparing for entry-level frontend, backend or full-stack interviews.

Module and package fees

Published module and programme fees in Indian rupees.

COMPLETE COURSE FEE **₹28,500**

COURSE MODULES & PATHWAYS	INR
Web Design	₹7,500
Advanced JavaScript & ECMAScript (ES6–ES10)	₹4,000
React, Redux & Redux Toolkit	₹8,500
Node.js & Express.js backend	₹8,500

DATABASE TRAINING

SQL and NoSQL database training is included with complete Full Stack enrolment.

Jaipur Engineers
Your One-Stop IT & Staffing Ecosystem

COURSE ENQUIRIES

Ask about the next batch

Contact Jaipur Engineers for the complete Full Stack course fee, duration and available batch timings.

CALL JAIPUR ENGINEERS

+91 70146 92039

EMAIL & WEBSITE

info@jaipurengineers.com

jaipurengineers.com

VISIT THE TRAINING CENTRE

122/228, Indra Path, Thadi Market

Mansarovar, Jaipur, Rajasthan 302020

SCAN FOR MAP

Full Stack course details

Node.js course details

MERN Stack / Full Stack with Node.js

Course content

MODULES 01–02

01 Web Design

Lessons & practice

- HTML semantics and forms
- CSS Grid and Flexbox
- Bootstrap, Tailwind CSS and Core JavaScript

HANDS-ON Build a responsive product form and check keyboard and input feedback.

02 Advanced JavaScript & ECMAScript

Lessons & practice

- ES6–ES10 syntax and modules
- Array and object transforms
- Fetch, promises and async/await

HANDS-ON Load products and show loading, empty and failed responses.

MERN Stack / Full Stack with Node.js

Course content

MODULES 03–04

03 React, Redux & Redux Toolkit

Lessons & practice

- React components, routes and hooks
- Controlled forms
- Redux Toolkit slices, selectors and async actions

HANDS-ON Use a Redux slice to manage catalogue filters and inventory state.

04 Node.js & Express.js

Lessons & practice

- Node.js modules and runtime
- Express routes and middleware
- Validation and structured errors

HANDS-ON Build an order endpoint and check its result when a required field is missing.

MERN Stack / Full Stack with Node.js

Course content

MODULES 05–06

05 MongoDB & database practice

Lessons & practice

- MongoDB documents and indexes
- SQL relationships and queries
- CRUD and consistency

HANDS-ON Store an order and verify a query that retrieves a customer's order history.

06 Integration and delivery

Lessons & practice

- End-to-end tests
- Permissions and runtime configuration
- Deployment and documentation

HANDS-ON Trace an order through React, Express and MongoDB and document how to run the project.