

# Architecture Overview

Your project is a **full-stack web application** with a **React frontend** and **Node.js backend**, connected to **MongoDB database**.

## 🔗 Servers Used

### Frontend Server (Vite)

- **Port:** 8080
- **URL:** http://localhost:8080
- **Purpose:** Serves the React application (what students see in browser)
- **Command:** npm run dev

### Backend Server (Express.js)

- **Port:** 3001
- **URL:** http://localhost:3001
- **Purpose:** Handles API requests and database operations
- **Command:** npm run server

## 🗄 Database: MongoDB Atlas

### What is MongoDB Atlas?

- **Cloud database service** (like Google Drive for data)
- **Free tier available** for students
- **Stores todos, user data, and application information**

### Why We Use It:

- **Persistent storage** - data doesn't disappear when you close the app
- **User-specific data** - each student sees only their own todos
- **Scalable** - can handle many users

## 🔌 APIs (Application Programming Interfaces)

### Authentication APIs

POST /api/auth/login - Student logs in  
GET /api/auth/verify - Check if student is still logged in  
POST /api/auth/logout - Student logs out

### Todo Management APIs

GET /api/todos - Get all todos for the student  
POST /api/todos - Create a new todo  
PUT /api/todos/:id - Update an existing todo  
DELETE /api/todos/:id - Delete a todo  
PATCH /api/todos/:id/toggle - Mark todo as complete/incomplete  
GET /api/todos/stats - Get todo statistics (total, completed, etc.)

### How APIs Work:

1. **Frontend** (React) sends request to **Backend** (Express)
2. **Backend** processes request and talks to **MongoDB**
3. **MongoDB** returns data to **Backend**
4. **Backend** sends response back to **Frontend**
5. **Frontend** displays the result to student

## 🔒 IP Address Modification

### What is IP Address?

- **Internet Protocol Address** - like a home address for your computer
- **Example:** 192.168.1.100 or 106.216.195.189

### Why We Modified IP Whitelist:

1. **Security:** MongoDB Atlas blocks unknown IP addresses by default
2. **Access Control:** Only whitelisted IPs can connect to the database
3. **Your IP:** 106.216.195.189 needed to be added to MongoDB Atlas

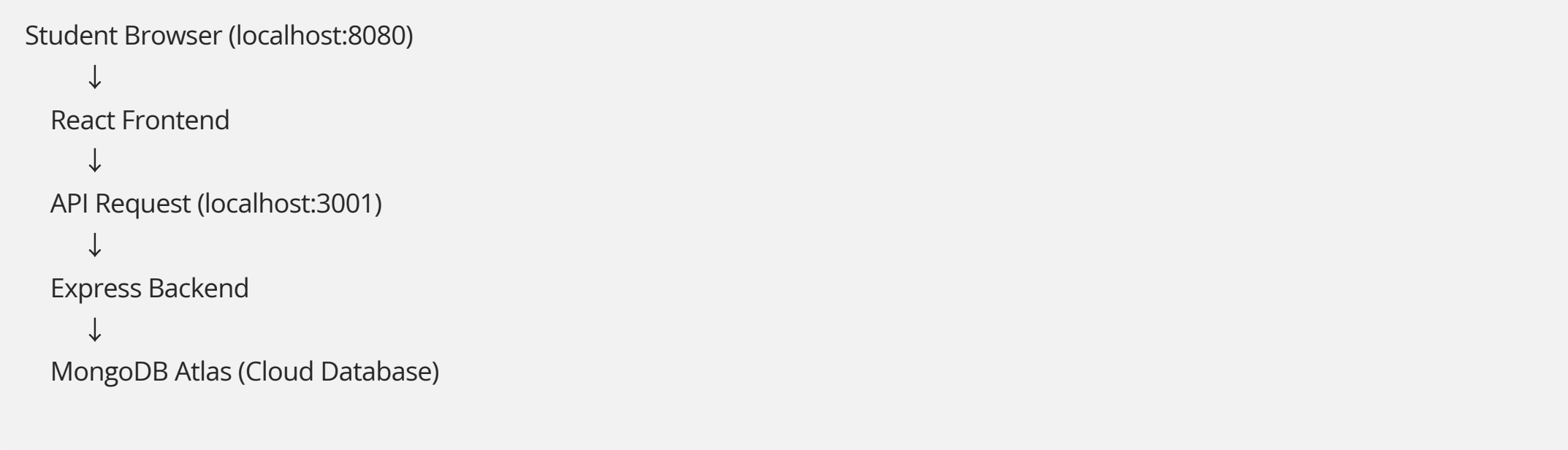
### The Problem:

- **MongoDB Atlas** said: "I don't know this IP address, so I'm blocking it"
- **Result:** Todos couldn't be saved to database
- **Solution:** Add your IP to MongoDB Atlas whitelist

### How to Fix:

1. Go to MongoDB Atlas dashboard
2. Click "Network Access"
3. Add IP: 0.0.0.0/0 (allows access from anywhere)
4. Or add their specific IP address

## 🔄 Data Flow



## 🔧 Key Technologies

### Frontend

- **React 18** - User interface framework
- **TypeScript** - Type-safe JavaScript
- **Tailwind CSS** - Styling
- **shadcn/ui** - Pre-built components

### Backend

- **Node.js** - JavaScript runtime
- **Express.js** - Web server framework
- **Mongoose** - MongoDB database connector
- **JWT** - Authentication tokens

### Database

- **MongoDB Atlas** - Cloud database
- **MongoDB** - NoSQL database system

## Why This Architecture?

### For Students:

- **Learn real-world development** - same tools used by companies
- **Understand full-stack** - frontend, backend, and database
- **Practice API development** - how apps communicate
- **Experience cloud services** - MongoDB Atlas

### For Teachers:

- **Scalable** - can handle many students
- **Secure** - each student has isolated data
- **Maintainable** - clean code structure
- **Educational** - covers modern web development concepts

## Setup Summary

1. **Clone project** → git clone <repository>
2. **Install dependencies** → npm install
3. **Create MongoDB Atlas account** → Get connection string
4. **Configure environment** → Create .env file with credentials
5. **Whitelist IP** → Add IP to MongoDB Atlas
6. **Run project** → npm run dev:full

