

# **“Artilane”**

## **PROJECT REQUIREMENTS**



### **TEAM MEMBERS**

Prashant Kumar Mishra (2100291530040)

Abhinav Kumar (2100291530002)

Arpit Gupta (2100291530011)

**BACHELOR OF TECHNOLOGY IN**  
**COMPUTER SCIENCE AND ENGINEERING CSE-AI&ML**

**To create an e-commerce website for artisans using React for the frontend, Django for the backend, and MongoDB as the database, we would need the following requirements:**

**Frontend (React):**

1. Node.js: Install Node.js to use npm (Node Package Manager) for managing dependencies and running scripts.
2. React: Set up a React application using Create React App or a similar boilerplate.
3. React Router: Implement React Router for handling navigation and routing within the application.
4. State Management: Utilize a state management library like Redux or React Context API to manage application state efficiently.
5. UI Framework: Choose a UI framework/library like Material-UI, Ant Design, or Bootstrap for building responsive and visually appealing components.
6. HTTP Requests: Use a library like Axios or Fetch API to handle HTTP requests to the backend API.
7. Authentication and Authorization: Implement user authentication and authorization features using libraries like Firebase, Auth0, or JWT (JSON Web Tokens).

**Backend (Strapi):**

- Install and set up Strapi, a headless CMS, for managing product data, user authentication, and other dynamic content.
- Define and create the necessary content types and fields in Strapi to store product information, user data, etc.
- Implement authentication and authorization mechanisms using Strapi's user management features.
- Configure APIs in Strapi to expose the required data endpoints for your frontend to consume.

**Payment Integration with Stripe:**

- Create an account with Stripe and obtain API keys.
- Install the Stripe SDK or library in your React project.
- Implement the necessary components and functions to handle the payment process, such as adding items to the cart, calculating the total price, and initiating the payment using Stripe's APIs.
- Handle the various payment events, such as successful payment, failed payment, and handling webhooks for verifying payment status.

**Testing and Deployment:**

- Test your website thoroughly, including frontend functionality, backend APIs, and payment processing.
- Set up a staging environment for final testing and bug fixing.
- Once everything is working correctly, deploy your React frontend to a hosting provider of your choice (e.g., Netlify, Vercel).
- Deploy your Strapi backend to a suitable server or hosting environment (e.g., Heroku, AWS, DigitalOcean).
- Configure any necessary environment variables and ensure proper communication between the frontend, backend, and Stripe.

**Budget Requirement:**

In this project no budget is required as most of the technologies are open source and other non open source technologies are freely available for learning purposes.