Online Bidding Application

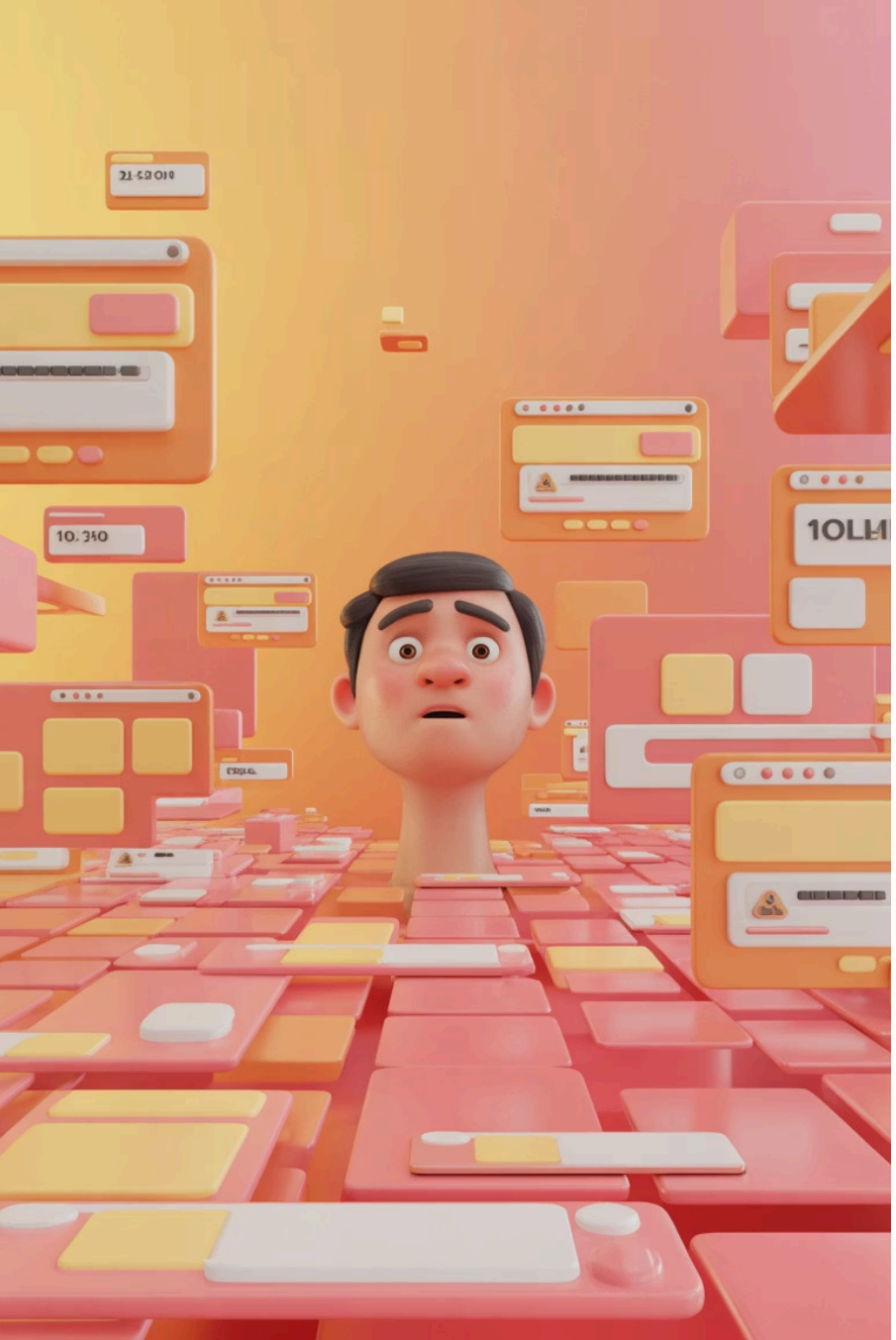
Srinivash | Roll No. - 23SUCA21

Full Stack Web App

Java, Spring Boot

Vite, JavaScript

DevOps Integrated



Introduction & Problem Statement

The current landscape of online marketplaces often lacks transparent and engaging bidding mechanisms. Users face challenges with real-time updates, secure transactions, and a streamlined experience for both buyers and sellers. Our project addresses these pain points by creating a robust and intuitive online bidding platform.

- Lack of real-time bidding transparency
- Complex product listing for sellers
- Inconsistent user experience across platforms
- Challenges in secure transaction management

Project Objectives

1

Secure & Scalable Backend

Develop a robust backend using Spring Boot for reliable API services and data management.

2

Intuitive User Interface

Create a responsive and user-friendly frontend with Vite for seamless bidding and product management.

3

Real-time Interaction

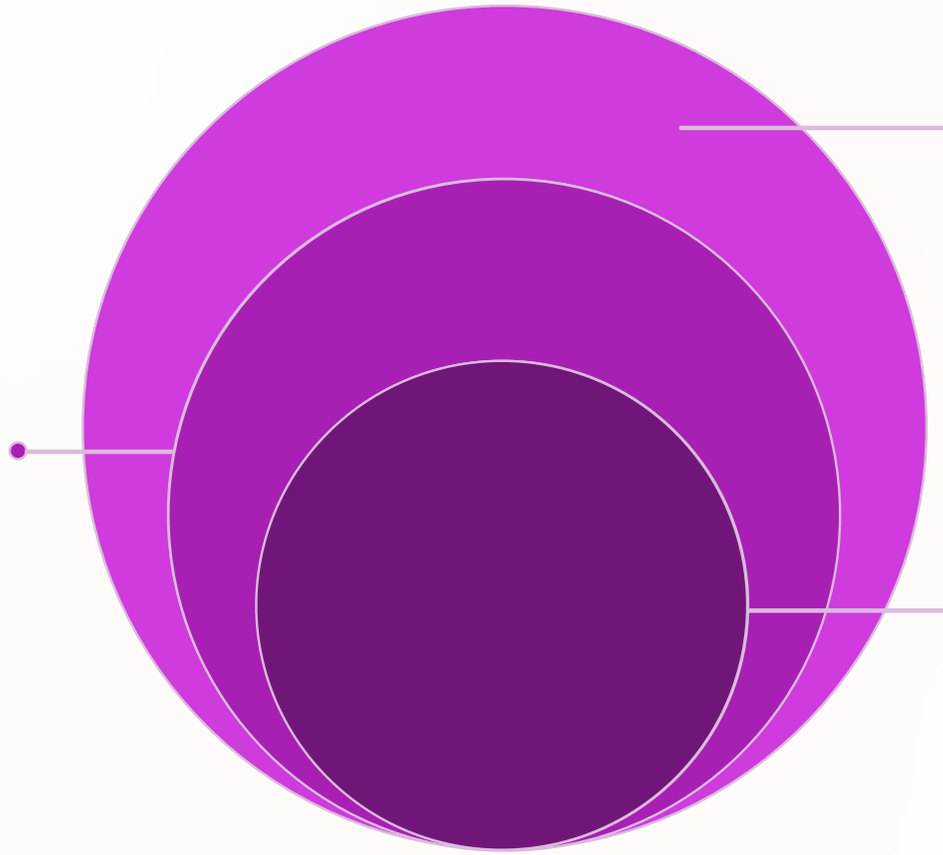
Implement features for live bidding updates and notifications to enhance user engagement.

4

Streamlined Deployment

Utilise Docker and cloud platforms (Render, Vercel) for efficient and continuous deployment.

System Architecture



Database

Persistent H2 storage

Backend

REST API, business logic, services

Frontend

User interface and API calls

Our application follows a standard three-tier architecture, ensuring clear separation of concerns and maintainability. The Frontend interacts with the Backend via REST APIs, which in turn communicates with the persistent H2 Database.

Comprehensive Technology Stack

We leveraged a modern and efficient technology stack across all layers of development, from backend services to frontend interfaces and robust DevOps practices, ensuring a high-performance and scalable solution.

Java

Core language for backend logic.

Spring Boot

Rapid application development framework.

Vite

Fast frontend development tooling.

Docker

Containerization for consistent environments.

Render

Backend cloud deployment.

Vercel

Frontend hosting and deployment.

Backend Development: Spring Boot REST APIs

The backend powers the entire application, handling data persistence, business logic, and secure communication with the frontend. Spring Boot significantly accelerated our development process.

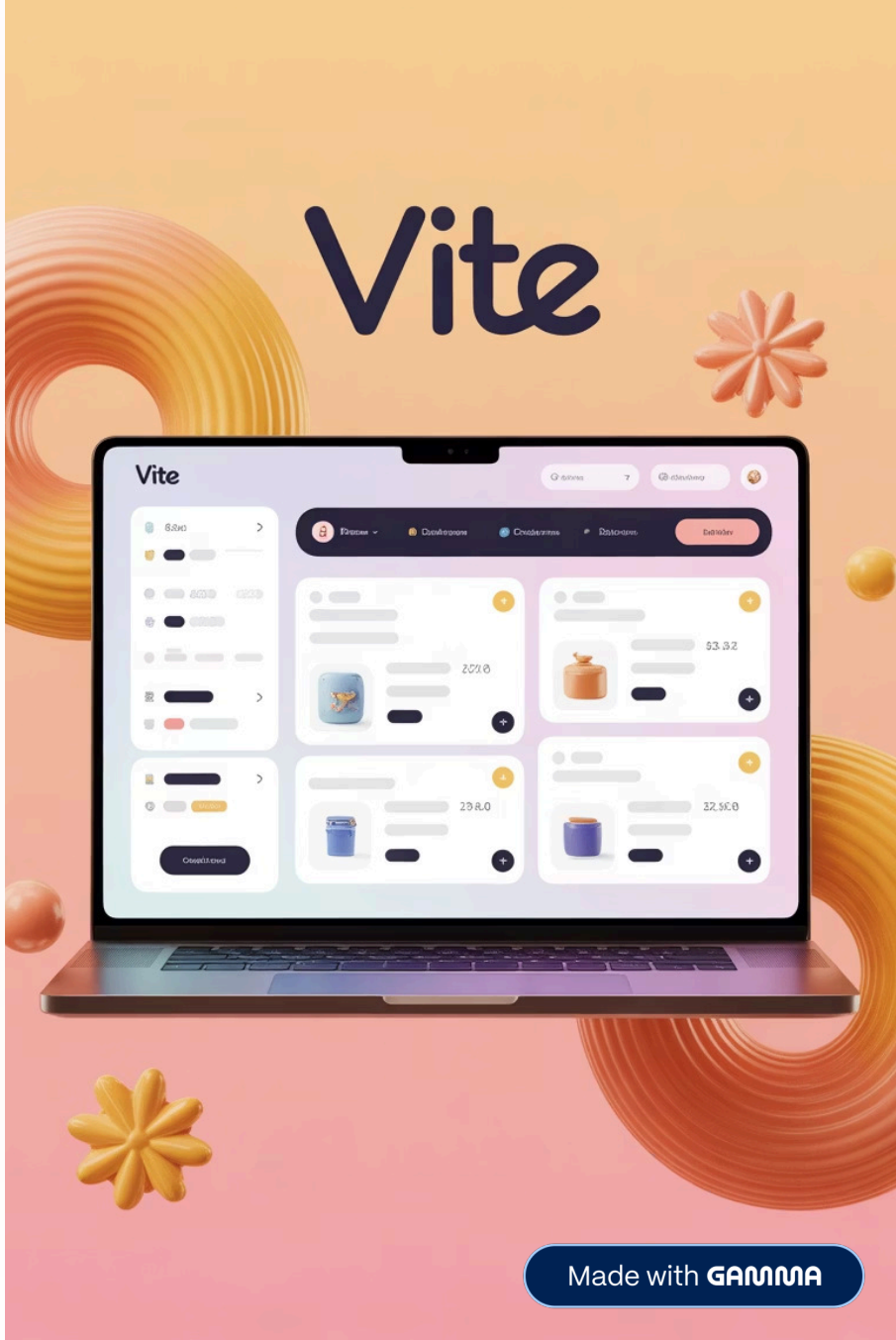
- **Java & Spring Boot:** Robust foundation for enterprise-grade applications.
- **Spring Data JPA:** Simplified data access and persistence with the H2 Database.
- **RESTful APIs:** Standardised communication for seamless frontend integration.
- **Maven:** Efficient dependency management and project build automation.

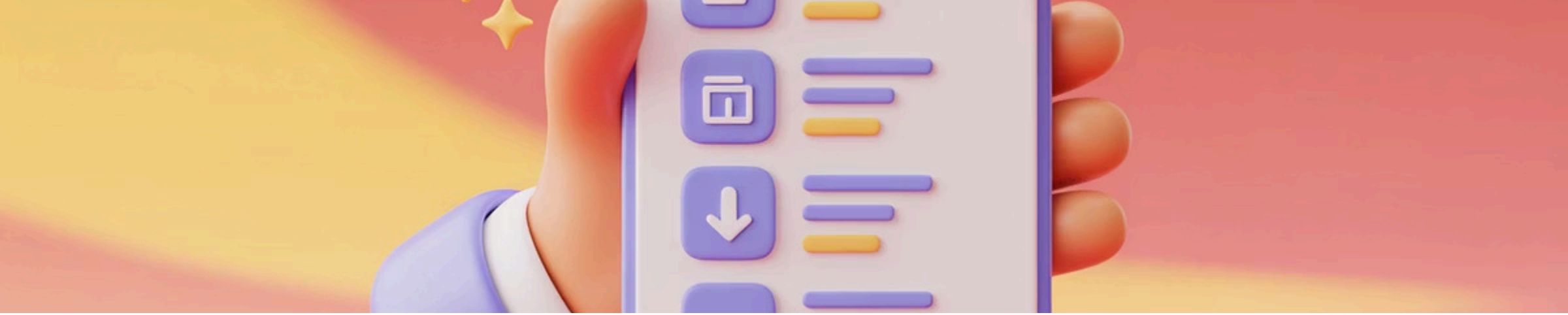
BACKEND APIS DATABASE

Frontend Development: Vite UI & API Integration

Vite provided an incredibly fast and efficient development experience for building our dynamic and interactive user interface. We focused on creating a smooth and engaging bidding journey.

- **Vite:** Next-generation frontend tooling for rapid development.
- **JavaScript, HTML, CSS:** Core web technologies for a rich user experience.
- **API Integration:** Seamless communication with Spring Boot backend for data exchange.
- **Responsive Design:** Optimised for various devices, ensuring accessibility.





Key Features & Functionalities

Product Upload & Management

Sellers can effortlessly list products with descriptions, images, and starting bid prices.

Dynamic Bidding System

Buyers can place bids in real-time, with automated notifications for higher bids.

CRUD APIs

Full Create, Read, Update, and Delete operations for products, bids, and user profiles.

User Authentication

Secure login and registration process for both buyers and sellers.

DevOps & Deployment

Our project embraced modern DevOps practices to ensure continuous integration, automated deployment, and code quality. This approach facilitates efficient development and reliable releases.

- **Docker:** Containerisation for consistent environments across development and production.
- **Render (Backend):** Cloud platform for deploying and hosting our Spring Boot APIs.
- **Vercel (Frontend):** High-performance hosting for our Vite-based user interface.
- **GitHub:** Version control for collaborative development and code management.
- **SonarQube/SonarCloud:** Automated static code analysis for quality assurance and bug prevention.
- **Postman:** API testing and documentation for robust backend services.

CI/CD AUTOMATION QUALITY

Challenges Faced & Future Enhancements

Challenges Faced

- Real-time bid synchronisation across clients.
- Optimising database queries for performance.
- Setting up a comprehensive CI/CD pipeline.
- Integrating secure payment gateways.

Future Enhancements

- Implement WebSockets for true real-time bidding.
- Introduce user-specific dashboards and analytics.
- Add a comprehensive review and rating system.
- Expand payment options and internationalisation.
- Develop a mobile application for broader reach.