**A MINI Project Report**

**On**

**Online Bidding System**



**Submitted by**

Mr. JASWANTH KUMAR PEDDINTI Mr. KOMAL KOUTILYA PENUMARTI

Regd. No: **21B91A05N5** Redg. No: **21B91A05P2**

Mr. HEMA SURYA SHANKHAR PENTAPATI Mr. SASI KANTH PONNADA

Regd. No: **21B91A05P1** Redg. No: **21B91A05Q0**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

S.R.K.R ENGINEERING COLLEGE



**CERTIFICATE**

This is to certify that this is a bonafide work on “Online Bidding System ” for and has been submitted by Mr. Jaswanth Kumar(21B91A05N5), Mr. Hema Surya Shankar(21B91A05P1),Mr. Komal Koutilya(21B91A05P2), Mr. Sasi Kanth(21B91A05Q0) as a MEAN STACK laboratory report, in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering , during the academic year 2023-24 . The candidate worked right under my Supervision and guidance.

**Lecturers In-Charge**

**Smt. P. Jahnavi**

**Sri. Ch Vinod Varma Head of the Department,**

**Smt. L. Divya** Dr. V. CHANDRA SHEKHAR

Assistant Professor, Head of the Department,

Department of CSE, Department of CSE

S.R.K.R Engineering College S.R.K.R. Engineering College, Bhimavaram. Bhimavaram.

# ACKNOWLEDGEMENT

I take immense pleasure in thanking **Dr. K. V. Mural Krishna Raju**, beloved principal of S.R.K.R Engineering College, Bhimavaram, **and Dr. V. Chandra Shekar**, esteemed Head of the Department (C.S.E), for having permitted me to carry out this Web Application Development Using Full Stack mini project work. I wish to express my deep sense of gratitude especially to my project Guide, **Smt. P. Jahnavi**, Assistant Professor, **Sri. Ch Vinod Varma**  , Assistant Professor, **Smt. L. Divya** Assistant Professor, for their guidance and useful suggestions, which helped me in completing the project work, on time. Finally, yet importantly, I would like to express my heartfelt thanks to my beloved parents, faculty, my friends/classmates for their help, and my wishes for the successful completion of this project.

Mr. JASWANTH KUMAR PEDDINTI

Regd. No: **21B91A05N5**

Mr. HEMA SURYA SHANKHAR PENTAPATI

Regd. No: **21B91A05P1**

Mr. KOMAL KOUTILYA PENUMARTI

Regd. No: **21B91A05P2**

Mr. SASI KANTH PONNADA

Regd. No: **21B91A05Q0**

TABLE OF CONTENTS

S. No CONTENTS Page No

1. ABSTRACT 5

1. MODULES 6

1. SOFTWARE REQUIREMENTS 7

1. HARDWARE REQUIREMENTS 7

1. TECHNICAL STACK 8 - 9

1. SAMPLE CODE 10 - 28

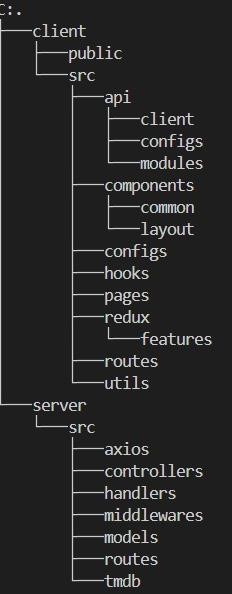
# SCREENSHOTS 29-31

# ABSTRACT

Online Bidding System: A Comprehensive Marketplace Abstract The Online Bidding System is a versatile digital marketplace that revolutionizes the traditional bidding process. It offers a seamless experience for users to log in, register products, quote bids, and negotiate prices. This platform is not just about transactions; it’s a community where users can communicate directly with potential bidders, fostering a transparent and efficient bidding environment. The key features are User Authentication, Product Registration, Bid Quotation, Price Negotiation, Direct Messaging. Meanwhile the primary advantages are Accessibility, Transparency, Efficiency, Community Building.

The Online Bidding System is designed to be intuitive, reliable, and secure, providing a comprehensive solution for the modern marketplace. Join us and elevate your bidding experience!

# MODULES



## SOFTWARE REQUIREMENTS

Text Editor:

Visual Studio Code, Sublime Text, Atom

Front End:

HTML, CSS, JavaScript, REACT JS

Back End:

MongoDB, Express.js, Node.js

Internet Browser:

Google Chrome, Mozilla Firefox, Microsoft Edge

## HARDWARE REQUIREMENTS

1. Processor-Intel Core i5
2. RAM- 4GB

## TECHNICAL STACK

HTML:

HTML (Hypertext Markup Language) is the standard markup language for creating web pages and web applications. It provides the structure and content of a webpage by using a system of tags and elements to define the various components such as headings, paragraphs, images, links, and forms. HTML documents are interpreted by web browsers to display the content to users. It forms the backbone of the web and is essential for building any web-based project.

CSS:

CSS (Cascading Style Sheets) is a stylesheet language used for describing the presentation and layout of HTML documents. It allows developers to style the visual appearance of web pages, including elements like fonts, colors, spacing, and positioning. CSS works by defining rules that specify how HTML elements should be displayed. It enables the separation of content from presentation, making it easier to maintain and customize the design of a website or application across multiple pages.

JavaScript (JS):

JavaScript is a versatile programming language commonly used in web development to add interactivity and dynamic behavior to websites. It runs on the client-side within web browsers, allowing developers to manipulate the content and structure of HTML and CSS in response to user actions. JavaScript can be used for tasks such as form validation, DOM manipulation, animations, and handling asynchronous requests. It is essential for creating modern, interactive web experiences.

PHP:

PHP (Hypertext Preprocessor) is a server-side scripting language designed for web development. It is widely used for building dynamic web pages and web applications, especially those that require interaction with databases. PHP code is executed on the server, generating HTML output that is then sent to the client's browser. It is commonly used in conjunction with databases like MySQL to create dynamic content, process form data, and perform other server-side tasks. PHP is a powerful tool for creating dynamic and data-driven websites.

MySQL:

MySQL is a popular open-source relational database management system (RDBMS) known for its reliability, performance, and ease of use. It is used to store and manage structured data in tables, where each table consists of rows and columns. MySQL supports SQL (Structured Query Language) for querying and manipulating data, making it suitable for a wide range of applications, from small websites to large-scale enterprise systems. It is commonly used in conjunction with PHP to create dynamic web applications that rely on database functionality. MySQL is an essential component of the LAMP (Linux, Apache, MySQL, PHP) stack, which is widely used for web development.

MongoDB:

MongoDB is a NoSQL database management system that provides high performance, scalability, and flexibility for storing and managing data in a document-oriented format.

Express.js:

Express.js is a minimalist web application framework for Node.js that simplifies the process of building web applications and APIs. It provides a robust set of features for routing, middleware, and handling HTTP requests and responses.

React.js:

React.js is a JavaScript library for building user interfaces, developed by Facebook. It allows developers to create reusable UI components and efficiently manage the state of an application, resulting in faster rendering and improved performance.

Node.js:

Node.js is an open-source, cross-platform JavaScript runtime environment that executes JavaScript code outside of a web browser. It is commonly used for building scalable serverside applications and APIs, leveraging asynchronous, event-driven architecture for high performance.

### SAMPLE CODES APP CODE .env

import AppStartUp from "./startup\_components/AppStartUp";

import AppCoreComponent from "./core\_components/AppCoreComponent";

function App(){

    return (

        <div>

            <AppStartUp />

            {/\* <AppCoreComponent /> \*/}

        </div>

    );

}

export default App;

### App\_StartUp.jsx

import { Grid, Stack } from "@mui/material";

import SignIn from "./SignIn";

import SignUp from "./SignUp";

import Header from "./Header";

import Banner from "./Banner";

import MiniProfile from "./MiniProfile";

import RoomCodePage from "./RoomCodePage";

import { useState } from "react";

function AppStartUP(){

    const [isLoggedIn,setIsLoggedIn]=useState(true);

    const [showSignIn,setShowSignIn]=useState(true);

    // a function to handle signin-signup hook in corresponding component files...

    function handleSign\_IN\_UP(needSignIn){

        if(needSignIn){

            setShowSignIn(true);

        }

        else{

            setShowSignIn(false);

        }

    }

    return (

        <Stack direction="column">

            <Header />

            <Banner />

            <br />

            <Grid container fullWidth height={"100px"}>

                <Grid item sx={{backgroundColor: "#eeeeee"}} md={7} padding={5} >

                    {/\* main room section... \*/}

                    <RoomCodePage />

                </Grid>

                <Grid item md={5} >

                    { /\* Mini Profile Section... \*/ }

                    {isLoggedIn ? <MiniProfile /> : (showSignIn ? <SignIn handleSign\_IN\_UP={handleSign\_IN\_UP} /> : <SignUp handleSign\_IN\_UP={handleSign\_IN\_UP} />)}

                </Grid>

</Grid>

        </Stack>

    );

}

export default AppStartUP;

### RoomCode.jsx

import { Paper,

    Typography,

    FormControl,

    FormControlLabel,

    FormLabel,

    Switch,

    Stack,

    Box,

    Button,

    Card,

    CardContent,

    CardActions,

    TextField,

    IconButton,

 } from "@mui/material";

import ShareIcon from '@mui/icons-material/Share';

import ContentCopyIcon from '@mui/icons-material/ContentCopy';

import { useState } from "react";

function RoomCodePage(){

    const [haveKey, setHaveKey]=useState(false);

    const [roomCode, setRoomCode]=useState(null);

    const [buttonText,setButtonText]=useState("Create Room");

    const [roomCreated,setRoomCreated]=useState(false);

    function handleCodeChange(event){

        setRoomCode(event.target.value);

    }

    function handleCreateRoomClick(event){

        if(haveKey)

        {

            // Write a Room Joining Code with Room Code in useState(roomCode)

        }

        else{

            // Write Room Creation Code..

            setButtonText("Join Room");

            setRoomCreated(true);

        }

    }

    return (

        <Paper elevation={4} sx={{padding: "20px",width: "360px"}}>

            <FormControl>

            <Stack direction={"column"}>

                <Stack direction={"row"}>

                    <Box>

                    <FormLabel><Typography component={"span"} variant="body1">I have a Room Joining Code</Typography></FormLabel>

                    <FormControlLabel sx={{marginLeft:1}}

                    control={<Switch checked={haveKey}

                    onChange={function(event,bool\_value) {

                        bool\_value ? setButtonText("Join Room") : setButtonText("Create Room");

                        setRoomCreated(false);

                        setHaveKey(bool\_value)}

                        } />}

                        label={haveKey ? "Yes" : "No"} />

                    </Box>

                </Stack>

                <Stack direction={"row"}>

                    {haveKey ? <TextField label={haveKey ? "Enter Room Code" : ""} onChange={handleCodeChange} size="small" value={roomCode} /> : null}

                    <Button size="small" variant="contained" color="secondary" onClick={handleCreateRoomClick}>

                        <Typography variant="h6">{buttonText}</Typography>

                    </Button>

                    {

                        roomCreated ? <Card sx={{display: "flex", height: 40,marginLeft:1}} >

                            <CardContent sx={{color: "#ff5722", backgroundColor: "#fafafa"}} ><Typography sx={{marginTop: -1}} variant="h6">ABCDEF</Typography></CardContent>

                            <CardActions><IconButton>{<ContentCopyIcon size="small" color="secondary" />}</IconButton></CardActions>

                            <CardActions><IconButton>{<ShareIcon size="small" color="secondary" />}</IconButton></CardActions>

                        </Card> : null

                    }

                </Stack>

            </Stack>

            </FormControl>

        </Paper>

    );

}

export default RoomCodePage;

### SignIn.jsx

import { Typography, Stack, FormControl, TextField, Button } from "@mui/material";

function SignIn(props){

    function redirectSignUP(){

        props.handleSign\_IN\_UP(false);

    }

    return (

        <Stack direction={"column"} sx={{border: "solid 1px"}} spacing={12} padding={1}>

            <FormControl sx={{paddingLeft:8, paddingRight:8}}>

                <Typography sx={{textAlign: "center"}} variant="h4" gutterBottom >Login</Typography>

                <TextField size="small" label="Email ID" type="email" required />

                <br />

                <TextField size="small" label="Password" type="password" required />

                <Button size="small" onClick={redirectSignUP}>Don't have an account, Create One</Button>

            </FormControl>

        </Stack>

    );

}

export default SignIn;

### SIGNUP.jsx

import { Typography, Stack, FormControl, TextField, Button, getButtonGroupUtilityClass } from "@mui/material";

function SignUp(props){

    function redirectSignIN(){

        props.handleSign\_IN\_UP(true);

    }

    return (

        <Stack direction={"column"} sx={{border: "solid 1px"}} spacing={12} padding={1}>

            <FormControl sx={{paddingLeft:8, paddingRight:8}}>

                <Typography sx={{textAlign: "center"}} variant="h4" gutterBottom >Sign UP</Typography>

                <TextField size="small" label="Email ID" type="email" required />

                <br />

                <TextField size="small" label="Password" type="password" required />

                <br />

                <TextField size="small" label="Confirm Password" type="password" required />

                <Button size="small" onClick={redirectSignIN}>Already have an account, Login</Button>

            </FormControl>

        </Stack>

    );

}

export default SignUp;

### Header.jsx

import { Stack, Typography, Button, ButtonGroup } from "@mui/material";

import HomeIcon from '@mui/icons-material/Home';

import DescriptionIcon from '@mui/icons-material/Description';

import SportsMartialArtsIcon from '@mui/icons-material/SportsMartialArts';

import PeopleIcon from '@mui/icons-material/People';

function Header(){

    return (

        <Stack direction={"column"} spacing={1}>

            <Typography variant="h3" gutterBottom sx={{color: "secondary.main", paddingLeft:"32px"}}>Code Arena</Typography>

            <Stack direction={"row"} spacing={5} pl={4} pr={4}>

                <ButtonGroup fullWidth orientation="horizontal" size="large" color="secondary" >

                    <Button startIcon={<HomeIcon />} >Home</Button>

                    <Button startIcon={<DescriptionIcon />} >References</Button>

                    <Button startIcon={<SportsMartialArtsIcon />} >Exercises</Button>

                    <Button startIcon={<PeopleIcon />} >About Us</Button>

                </ButtonGroup>

            </Stack>

            <hr />

        </Stack>

    );

}

export default Header;

### Profile.jsx

import {Stack, Card, CardContent, CardMedia, CardActions, Typography} from "@mui/material";

function MiniProfile(){

    return (

        <Stack direction={"column"}>

            <Card sx={{padding: 2, paddingLeft:3, display: "flex"}}>

                <CardMedia component={"img"} sx={{width:200, height:200, borderRadius:"50%"}} alt="profile-photo" src="https://avatars.githubusercontent.com/u/122853435?v=4" />

                <CardContent>

                    <Typography variant="h5" aria-label="name" >Komal Koutilya Penumarti</Typography>

                </CardContent>

                <CardActions></CardActions>

            </Card>

        </Stack>

    );

}

export default MiniProfile;

AppCore.jsx

import { Stack, Grid } from "@mui/material"

import Header from "./Header";

import CodeArea from "./CodeArea";

import QuestionArea from "./QuestionArea";

import TestCaseSubmitArea from "./TestCaseSubmitArea";

function AppCoreComponent(){

    return (

        <Stack spacing={0} padding={0} margin={0} >

            <Header />

            <hr />

            <Stack spacing={2}>

            <Grid container>

                <Grid item md={4.8} sx={{borderRight: "solid 2px #aaaaaa"}}>

                    <QuestionArea />

                </Grid>

                <Grid item md={7.2}>

                    <CodeArea />

                    <TestCaseSubmitArea />

                </Grid>

            </Grid>

            </Stack>

        </Stack>

    );

}

export default AppCoreComponent;

### Header.jsx

import { Stack, Typography, Grid, ButtonGroup, Button } from "@mui/material";

function Header(){

    return (

        <Stack direction={"row-reverse"}>

        <Grid container>

            <Grid item md={10}>

                <Typography variant="h4">1037 Substring with Concatenation of All Words</Typography>

            </Grid>

            <Grid item md={2}>

            <ButtonGroup>

                <Button>1</Button>

                <Button>2</Button>

                <Button>3</Button>

                <Button>4</Button>

                <Button>5</Button>

            </ButtonGroup>

            </Grid>

        </Grid>

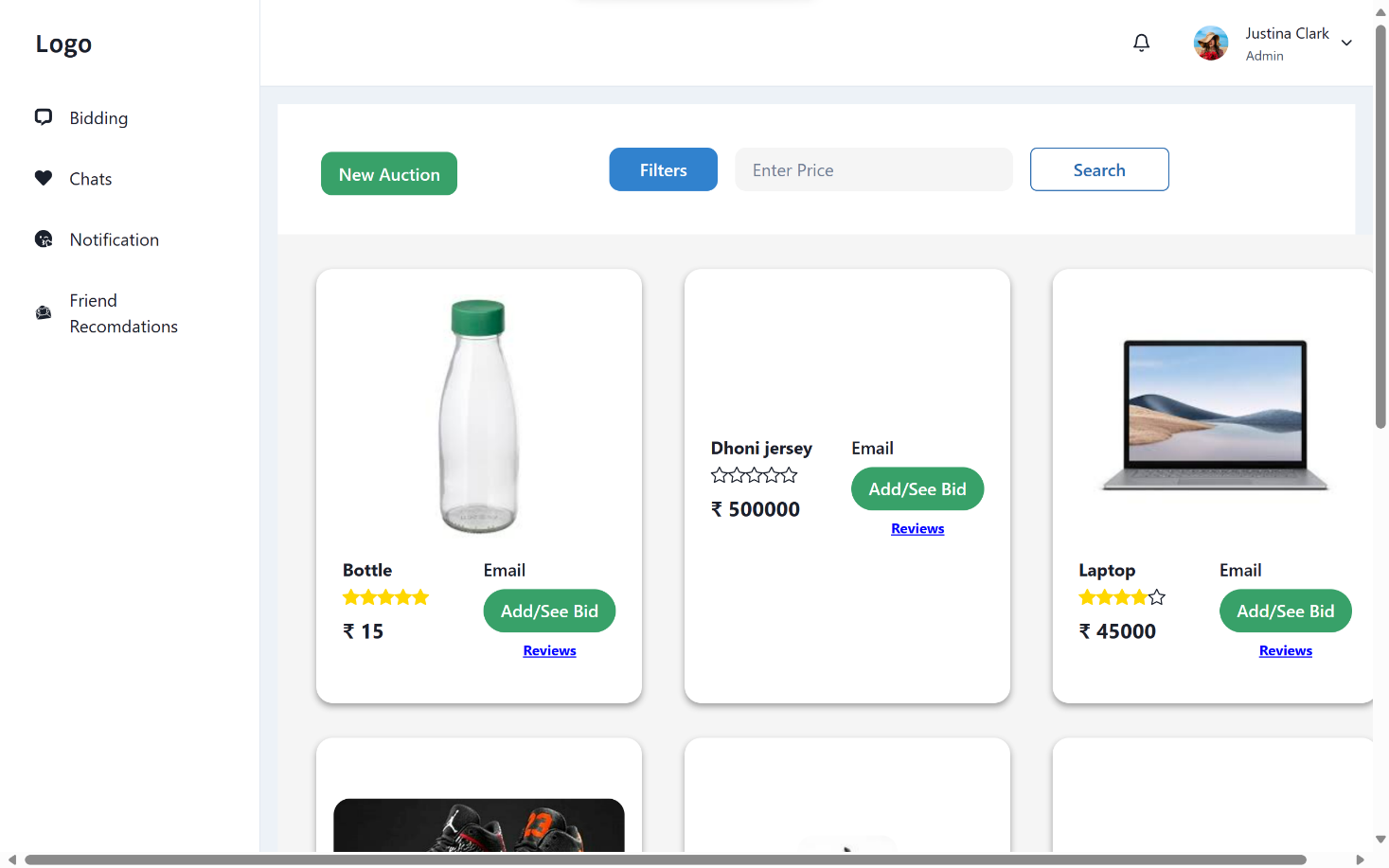
        </Stack>

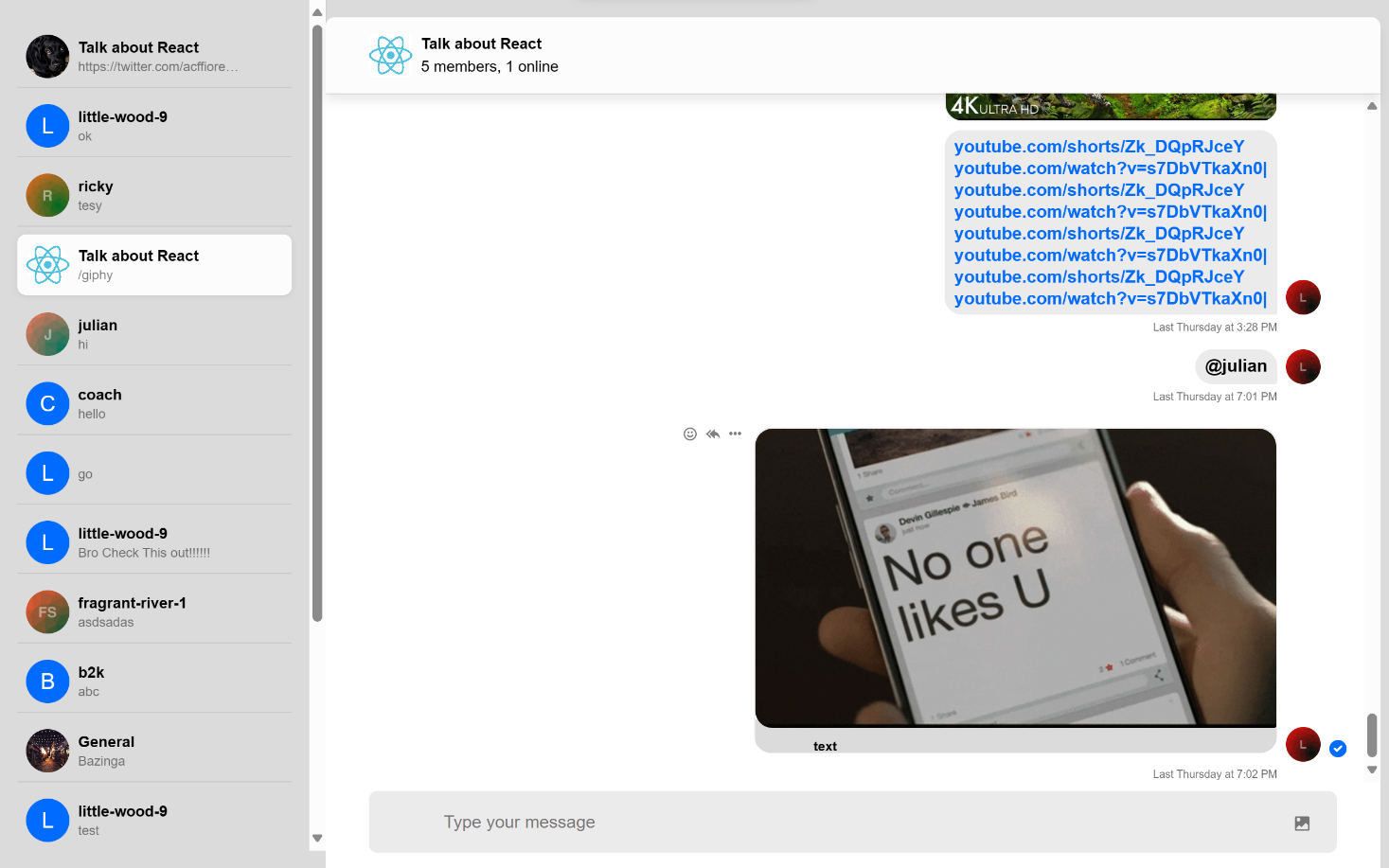
    );

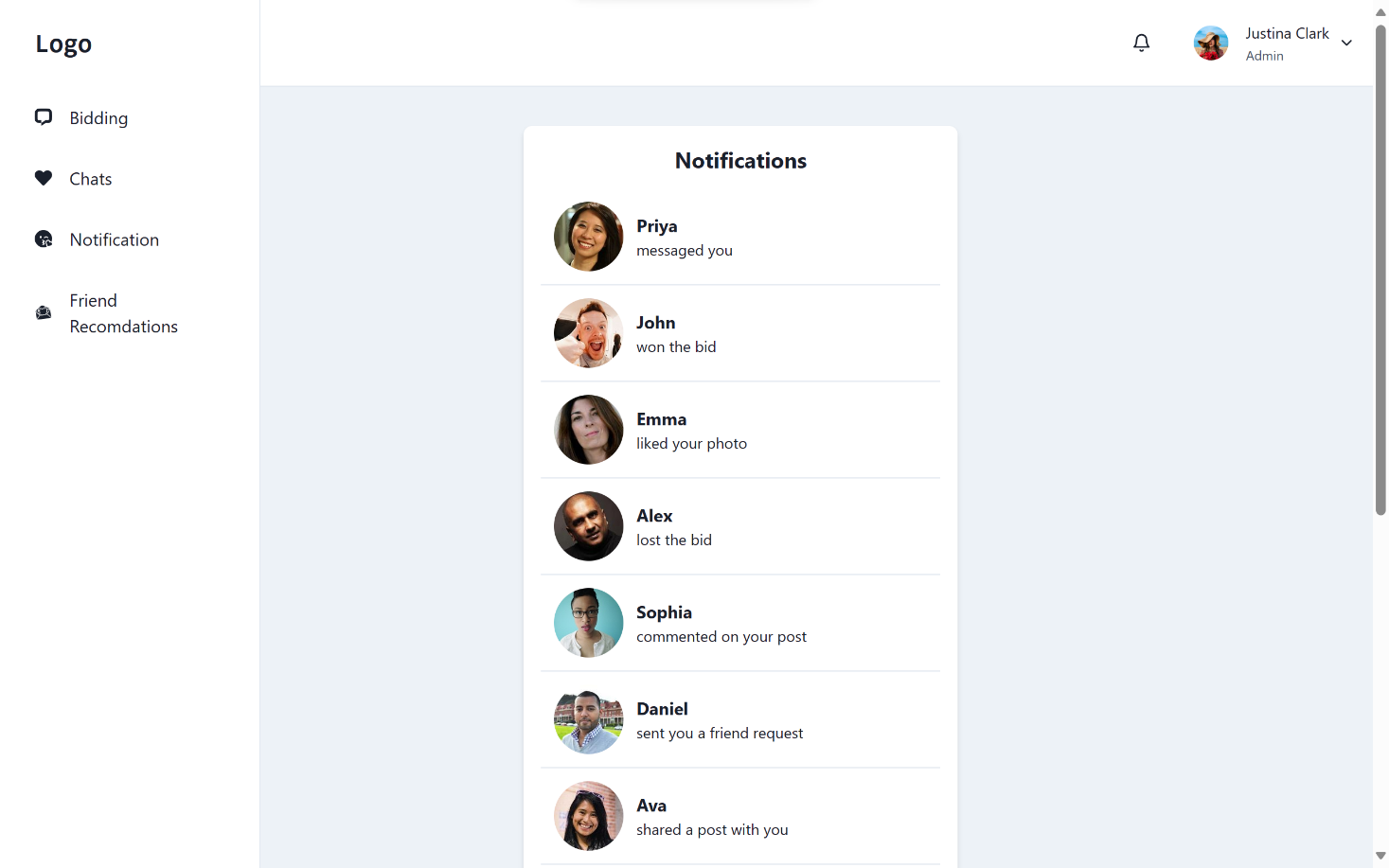
}

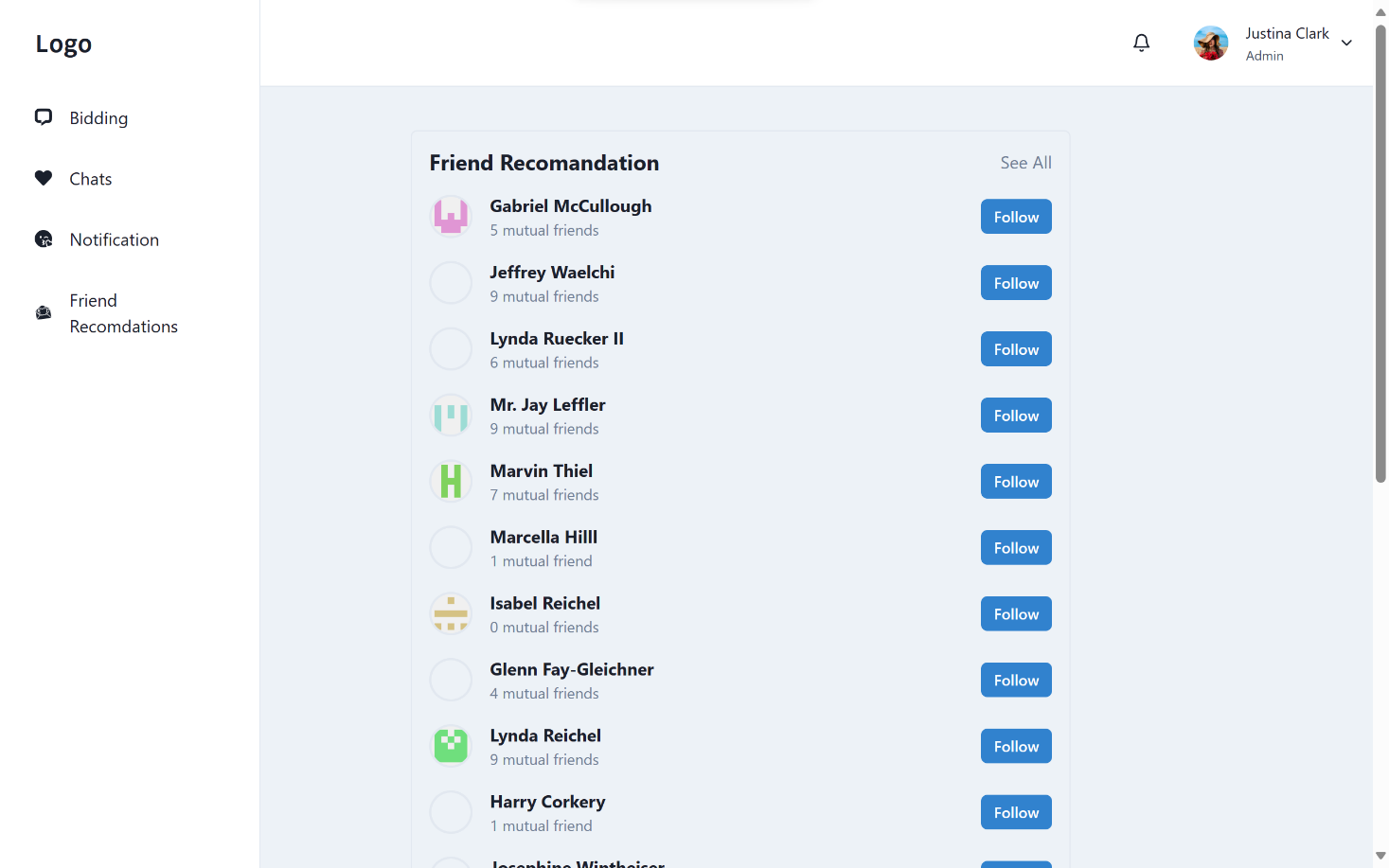
export default Header;

### SCREENSHOTS









### The project is deployed at: [Vite + React (weblink-three.vercel.app)](https://weblink-three.vercel.app/fr)