CS 261 | SOFTWARE DEVELOPMENT

W-IMs

An event management system

Table of Content

INTRODUCTION

Shouvik Ahmed Antu

Gregory Callahan

Douglas Li

Malie Heine

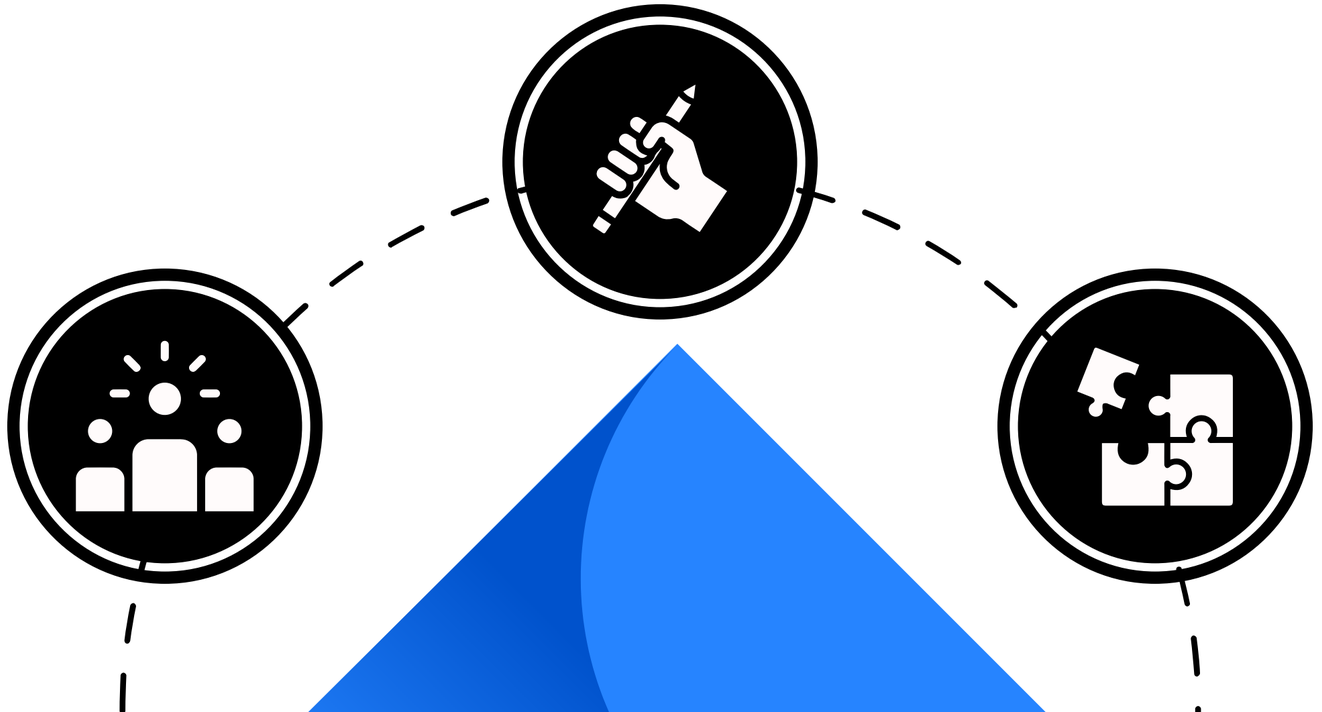
GOALS AND OBJECTIVES

Objective n° 1

Have a Final Project

Objective n° 2

Objective n° 3



GOALS AND OBJECTIVES

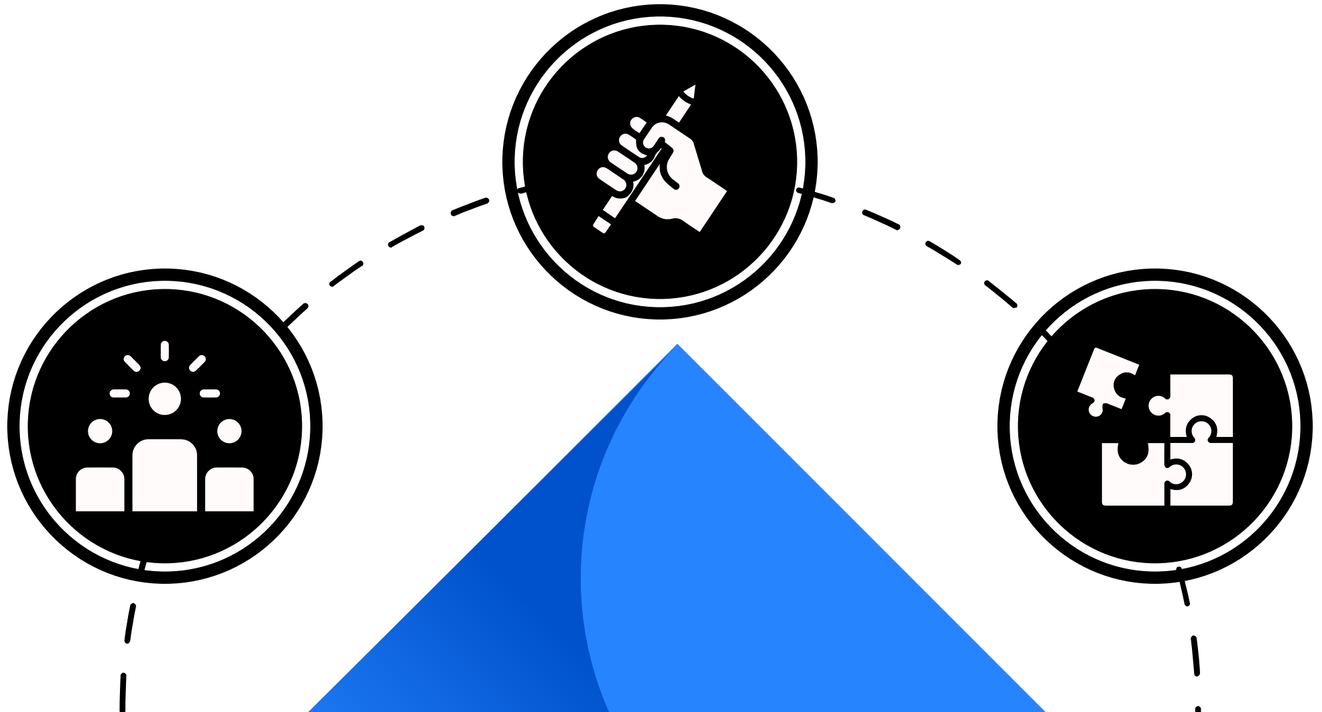
Objective n° 1

Have a Final Project

Objective n° 2

Have a Working Final Project

Objective n° 3



GOALS AND OBJECTIVES

Objective n° 1

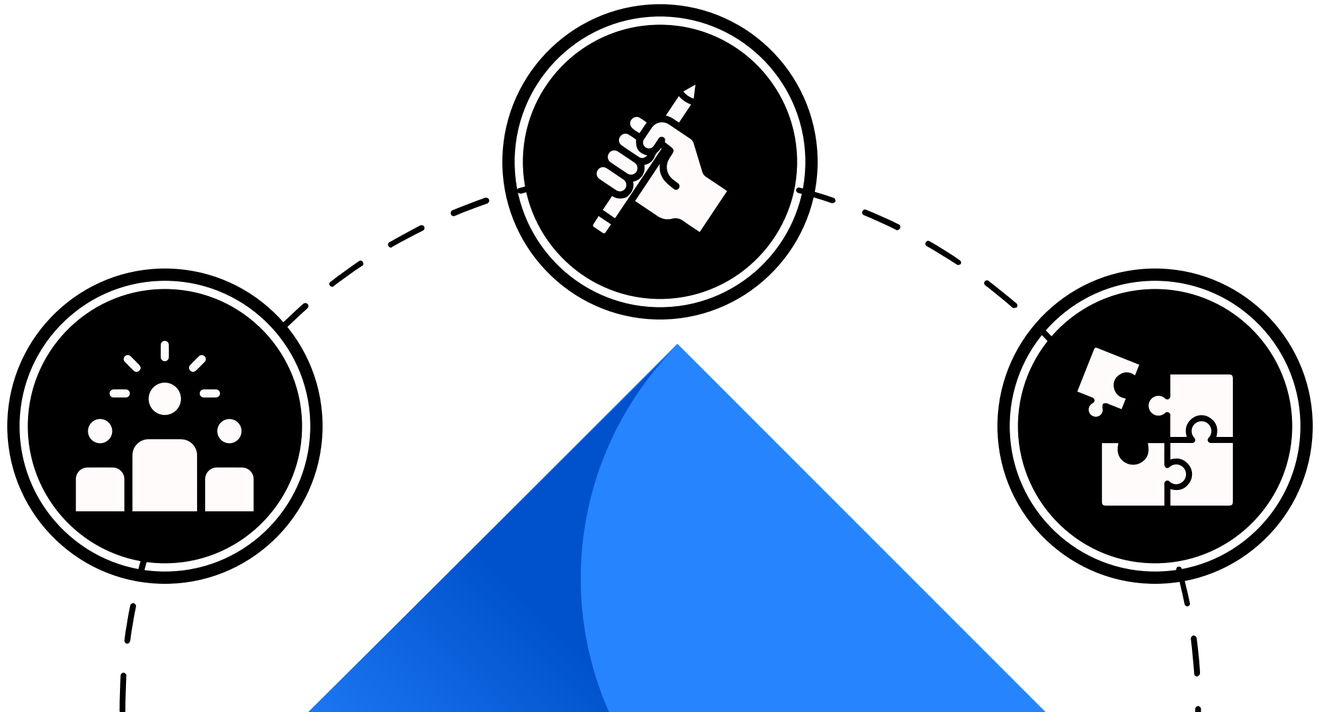
Have a Final Project

Objective n° 2

Have a Working Final Project

Objective n° 3

Have a complete final Project



GOALS AND OBJECTIVES

Objective n° 1

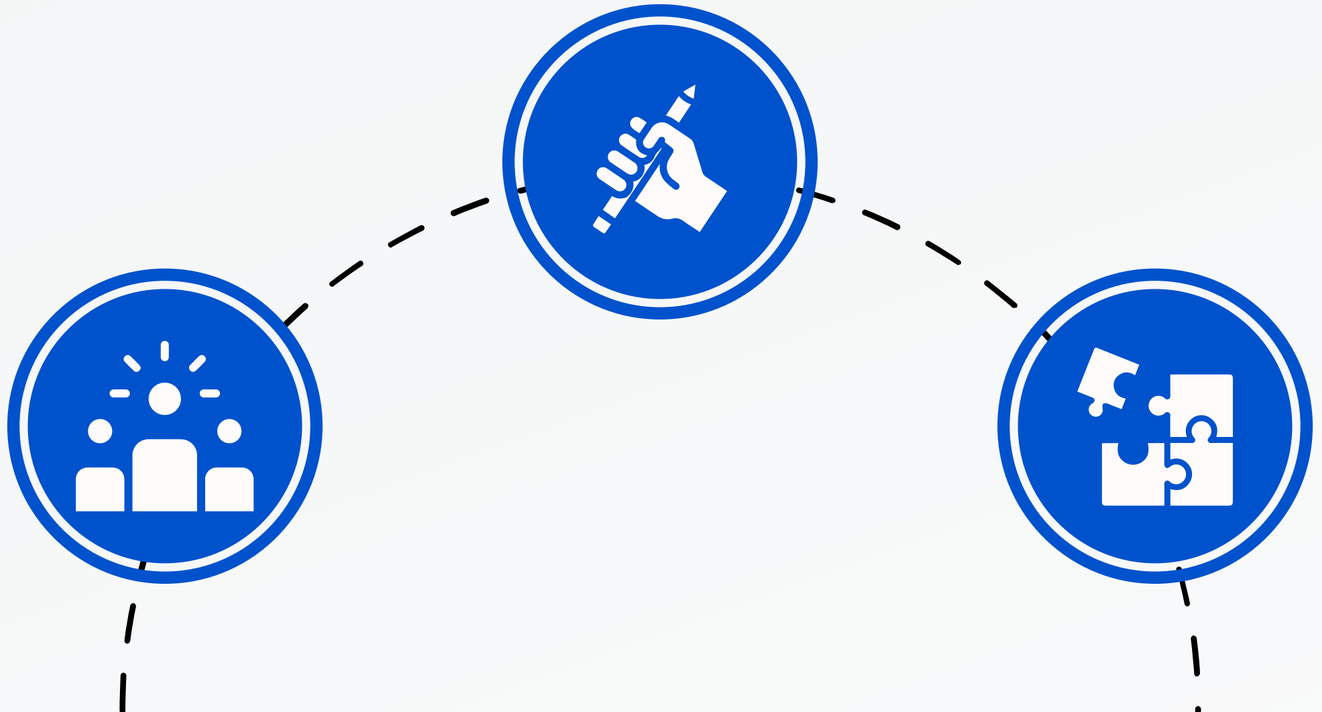
Something Useful. Can be used in real-life.

Objective n° 2

Learn How software development actually works

Objective n° 3

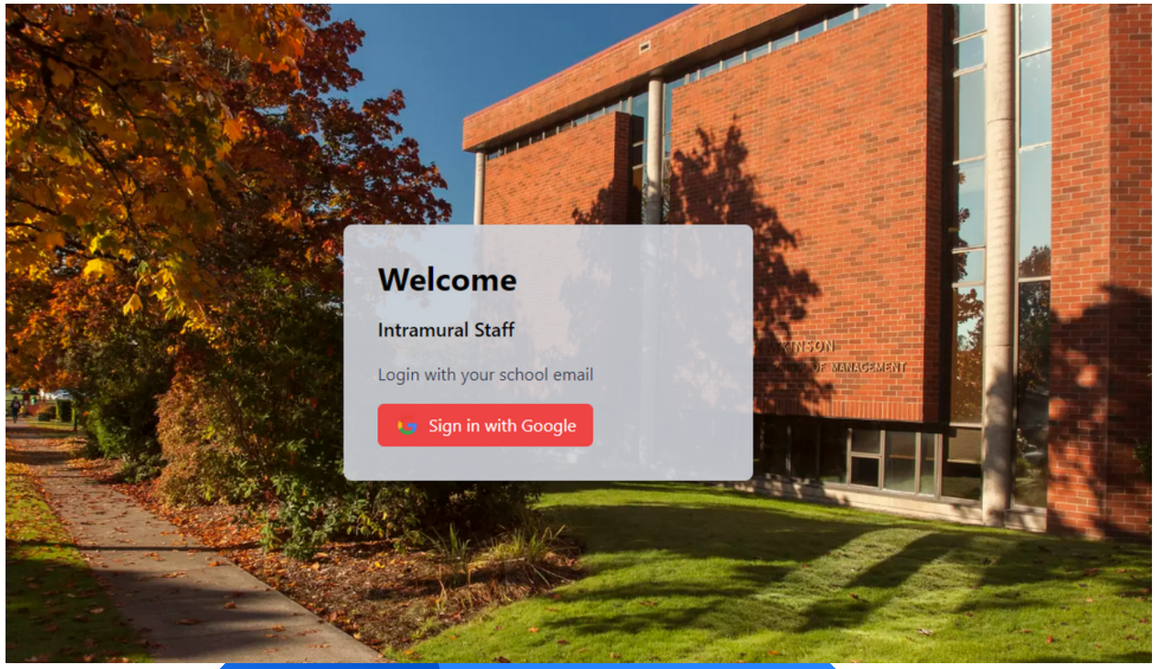
Have a good & working final Project



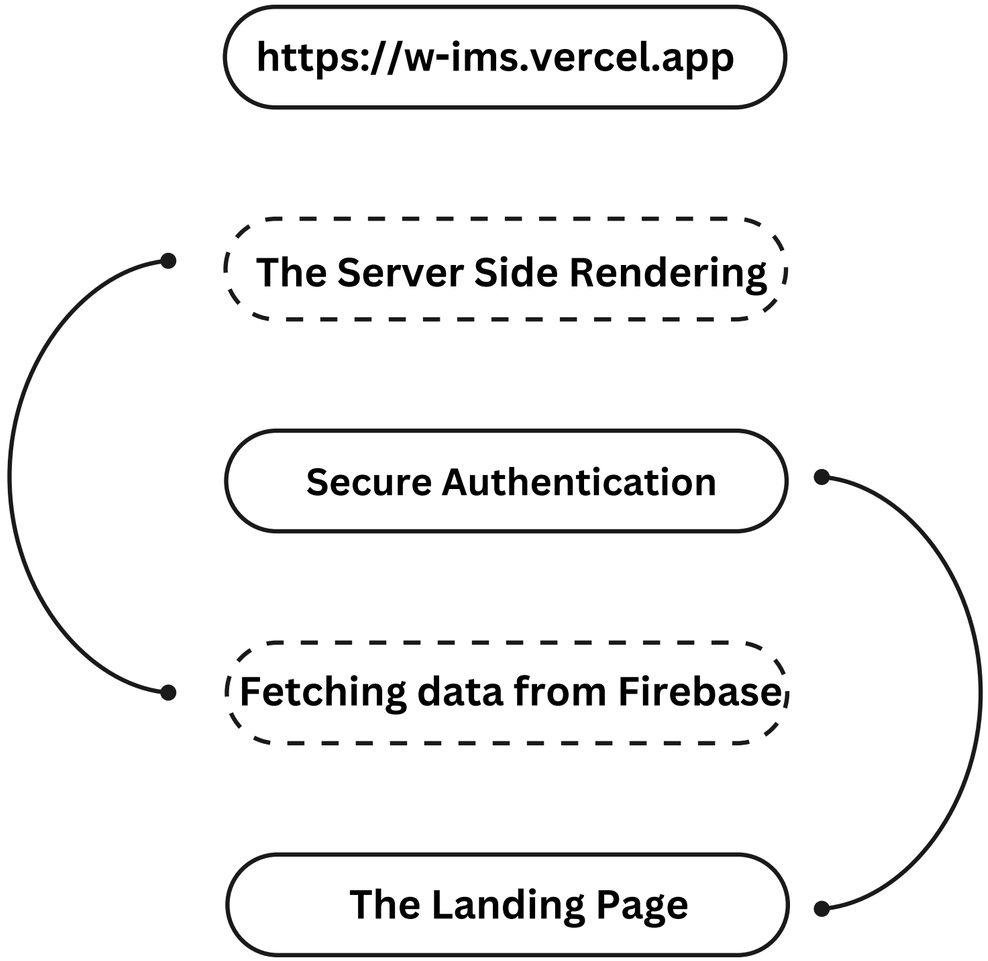
THE FULL TOUR

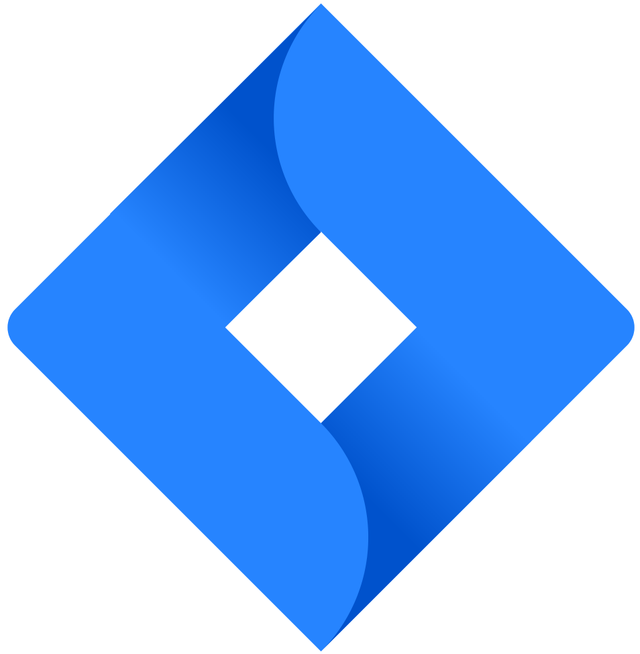
Existing System

The Better One



How it Works



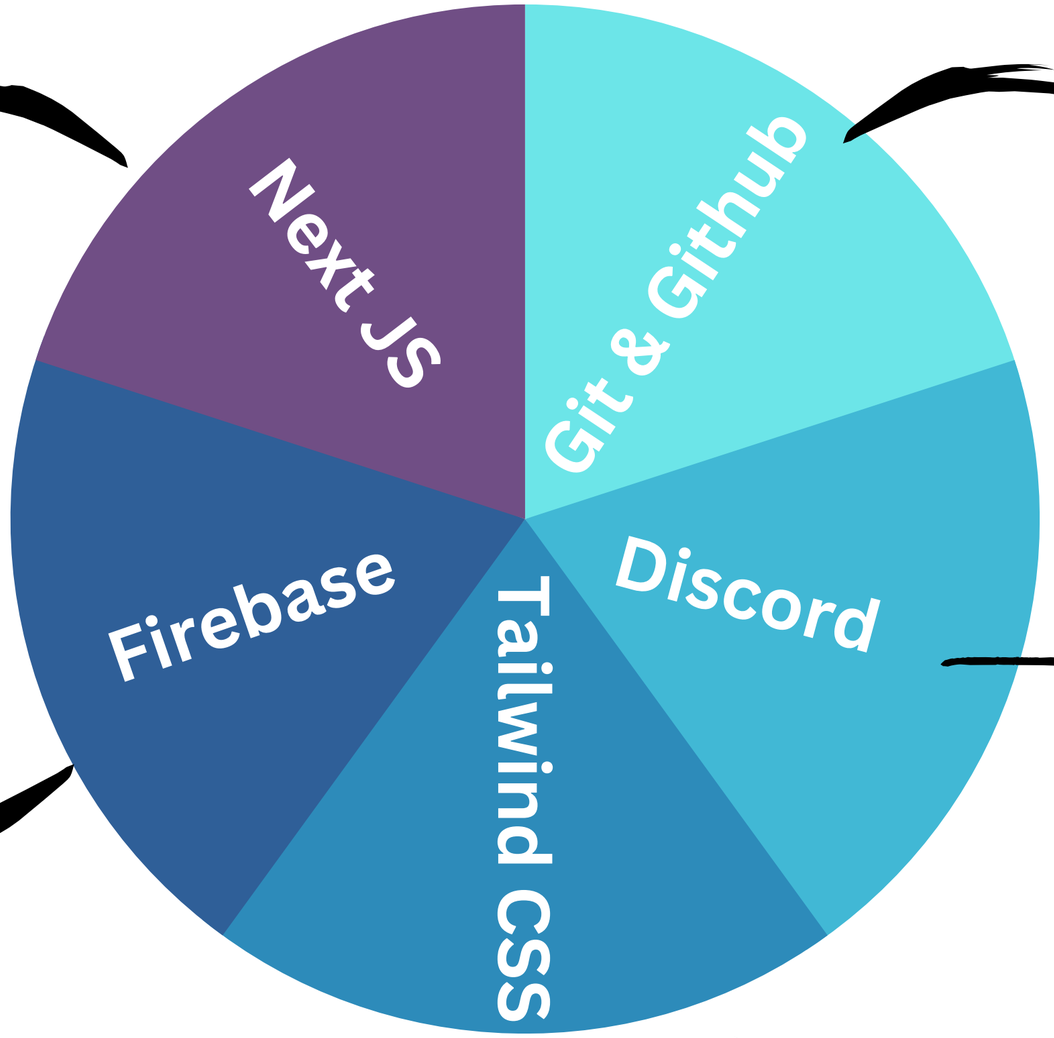


Built with a Plan

The Magic Behind it

Tech stacks & more


Next JS
Git & Github
Firebase
Discord
Tailwind CSS



Front-end

Management &
Version Control

Backend &
Database

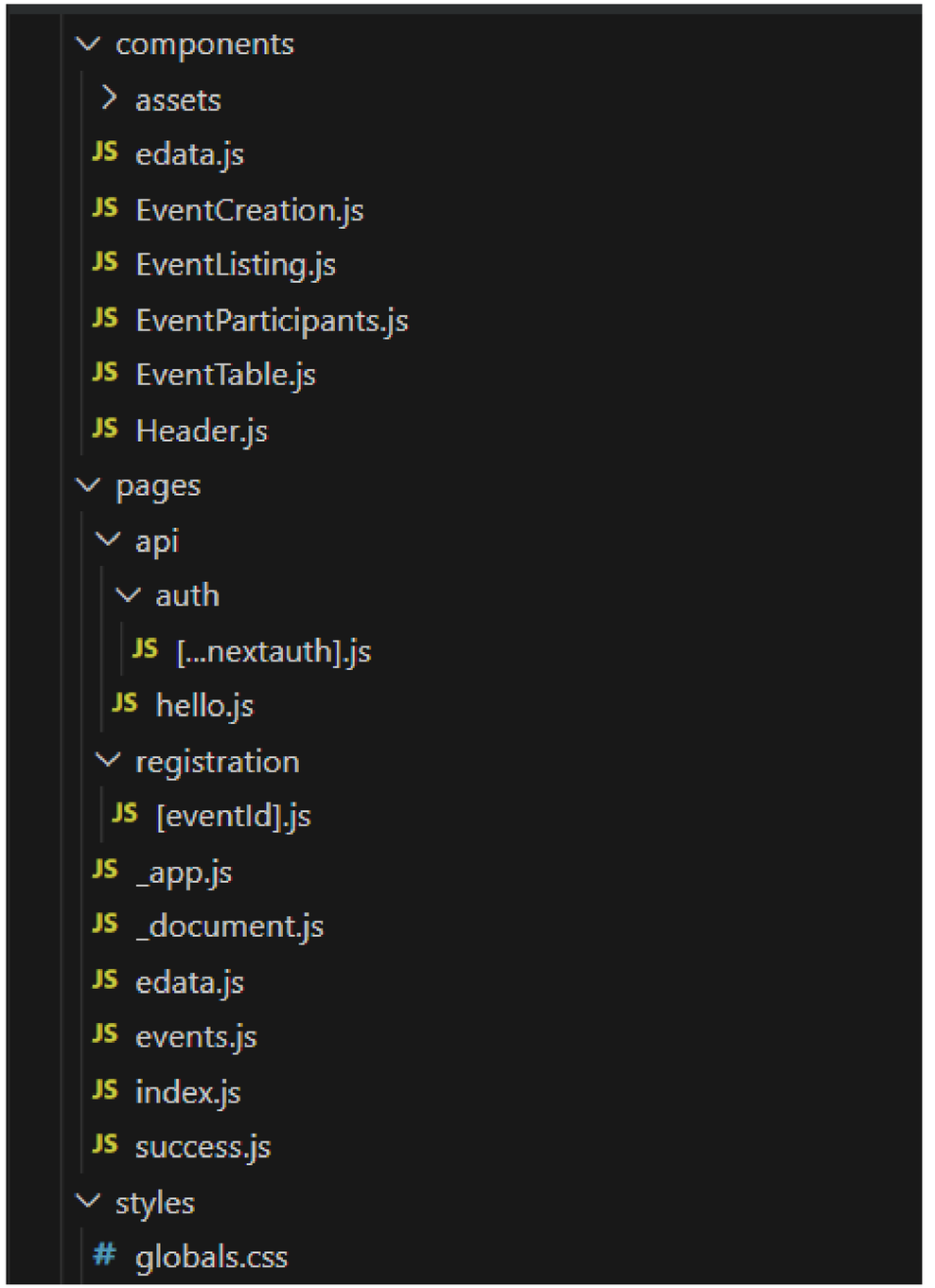
Communication

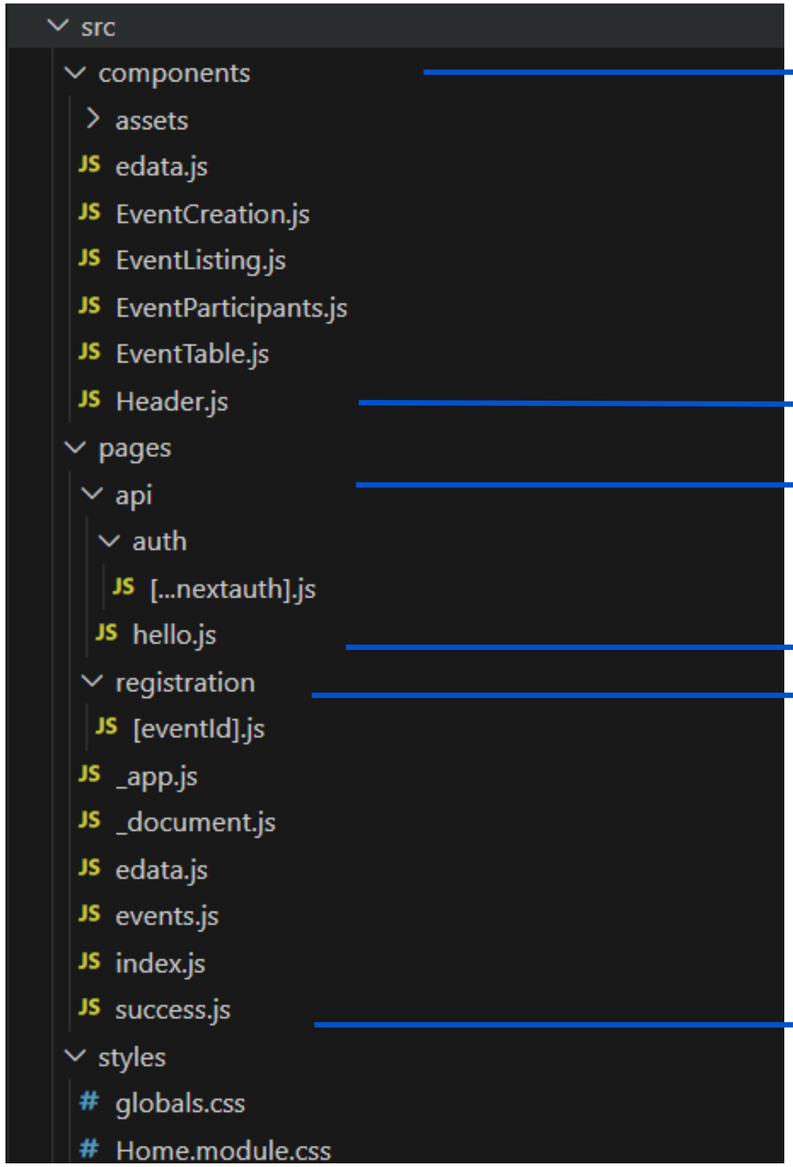
Design


```jsx
const handleLogin = (userData) => {
  setUser(userData);
  localStorage.setItem("user", JSON.stringify(userData)); // Store user in localStorage
};

const signIn = () => {
  signInWithPopup(auth, provider)
    .then((result) => {
      handleLogin(result.user); // call onLogin with user data
    })
    .catch((error) => alert(error.message));
};

return (
  <>
    {user ? (
      <div className="bg-container max-h-screen">
        <EventCreation user={user} />
      </div>
    ) : (
      <div className="bg-cover bg-center bg-container">
        <div className=" min-h-screen flex items-center justify-center">
          <div className="bg-gray-300 bg-opacity-90 p-8 rounded-lg shadow-lg max-w-sm w-full">
            <h1 className="text-3xl font-bold mb-4">Welcome</h1>
            <h2 className="text-lg font-medium mb-4">Intramural Staff</h2>
            <p className="text-gray-700 mb-4">
              Login with your school email
            </p>
```

- **Enhanced performance**
- **Lazy loading(Dynamic)**
- **Security is of the highest priority.**
- **Static site generation**

- **Clean File Structure**
- **Easy Navigation**
- **Secure API end-points**





Component -based design

Api & Authentication

Easy Navigation

NextJs Authentication via Google sign in

```
import NextAuth from "next-auth/next";
import GoogleProvider from "next-auth/providers/google";


export default NextAuth({
    providers: [
        GoogleProvider({
            clientId: process.env.GOOGLE_CLIENT_ID,
            clientSecret: process.env.GOOGLE_CLIENT_SECRET,
        }),
    ],
    callbacks: {
        async signIn({ user}) {
        if(user.email === "sahmed@willamette.edu" || user.email === "hcheng@willamette.edu"
        email==="qnottage@willamette.edu" || user.email === "gcallahan@willamette.edu"){
            const isAllowedToSignIn = true
            if (isAllowedToSignIn) {
                return true
              } else {
                return alert("Unauthorized")
              }
        }

        }
      },

    secret : process.env.NEXTAUTH_SECRET
});
```

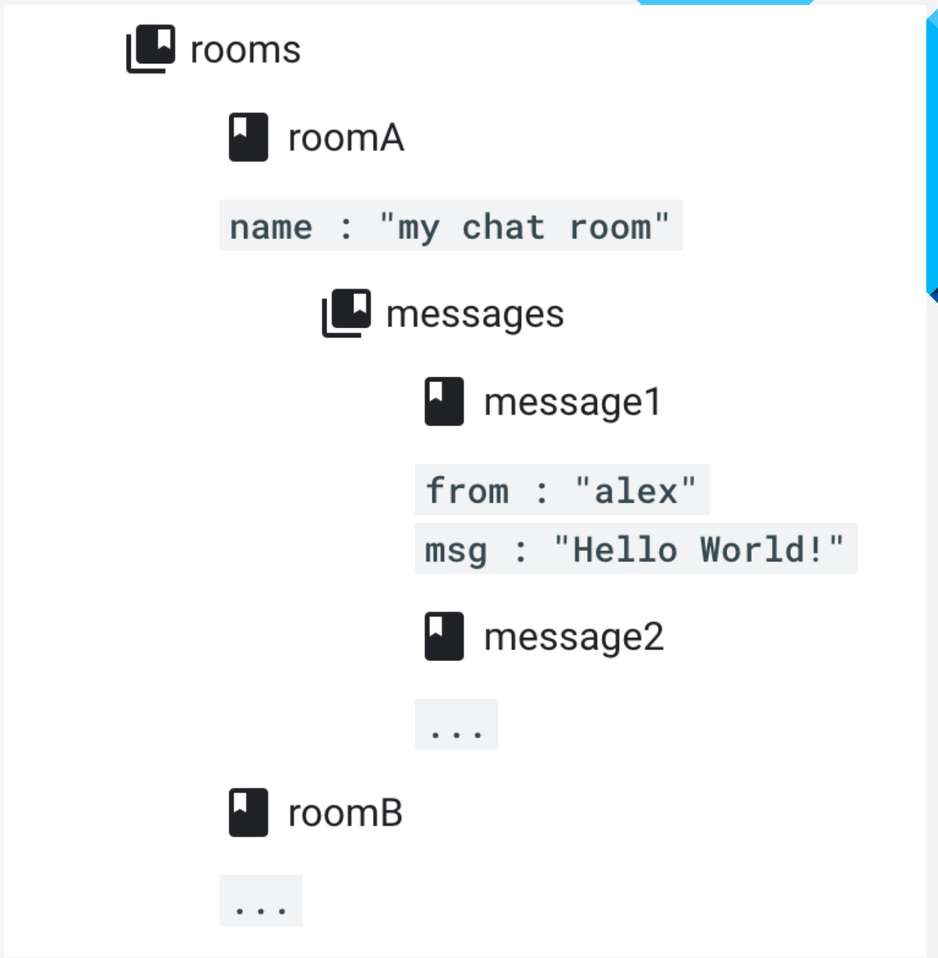
Firestore

- **Flexible Data Model**
- **Scalable Database**

Firebase

Cloud Firestore Data Model

Flexible data Model



Cloud Firestore Data Model

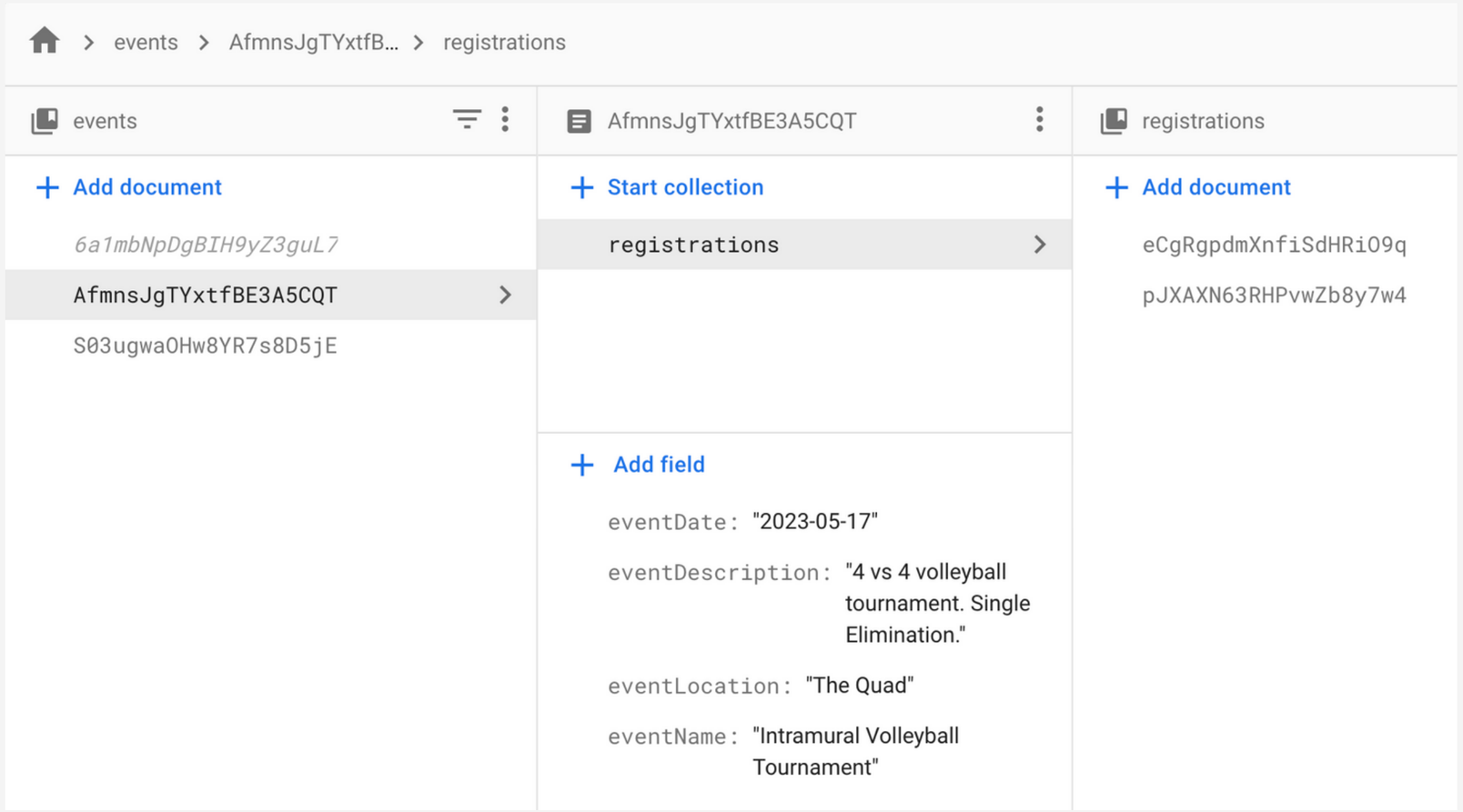
Flexible data Model

```
const EventParticipants = ({ ParentDocId }) => {
    const participantsRef = collection(db, "events", ParentDocId, "registrations")
    const [participants, setParticipants] = useState([])

    useEffect(() => {
        const participantscollection = onSnapshot(participantsRef, (snapshot) => {
            const newParticipants = snapshot.docs.map((doc) => ({
                id: doc.id,
                ...doc.data(),
            }));
            setParticipants(newParticipants);
        });

        return () => participantscollection
    }, []);
```

Cloud Firestore Data Model

Flexible data Model

Scalable

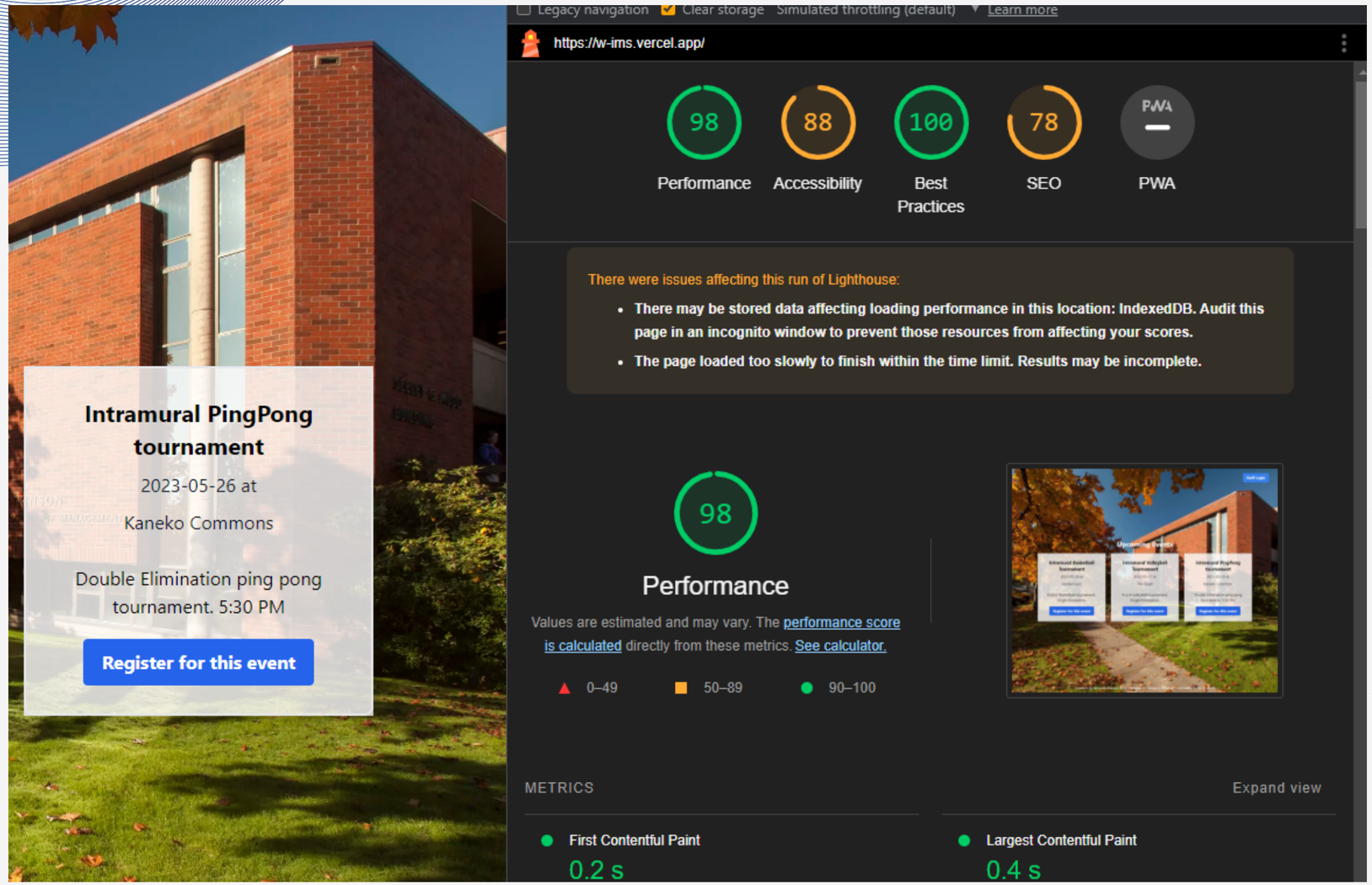


Cloud Firestore Data Model

Flexible data Model

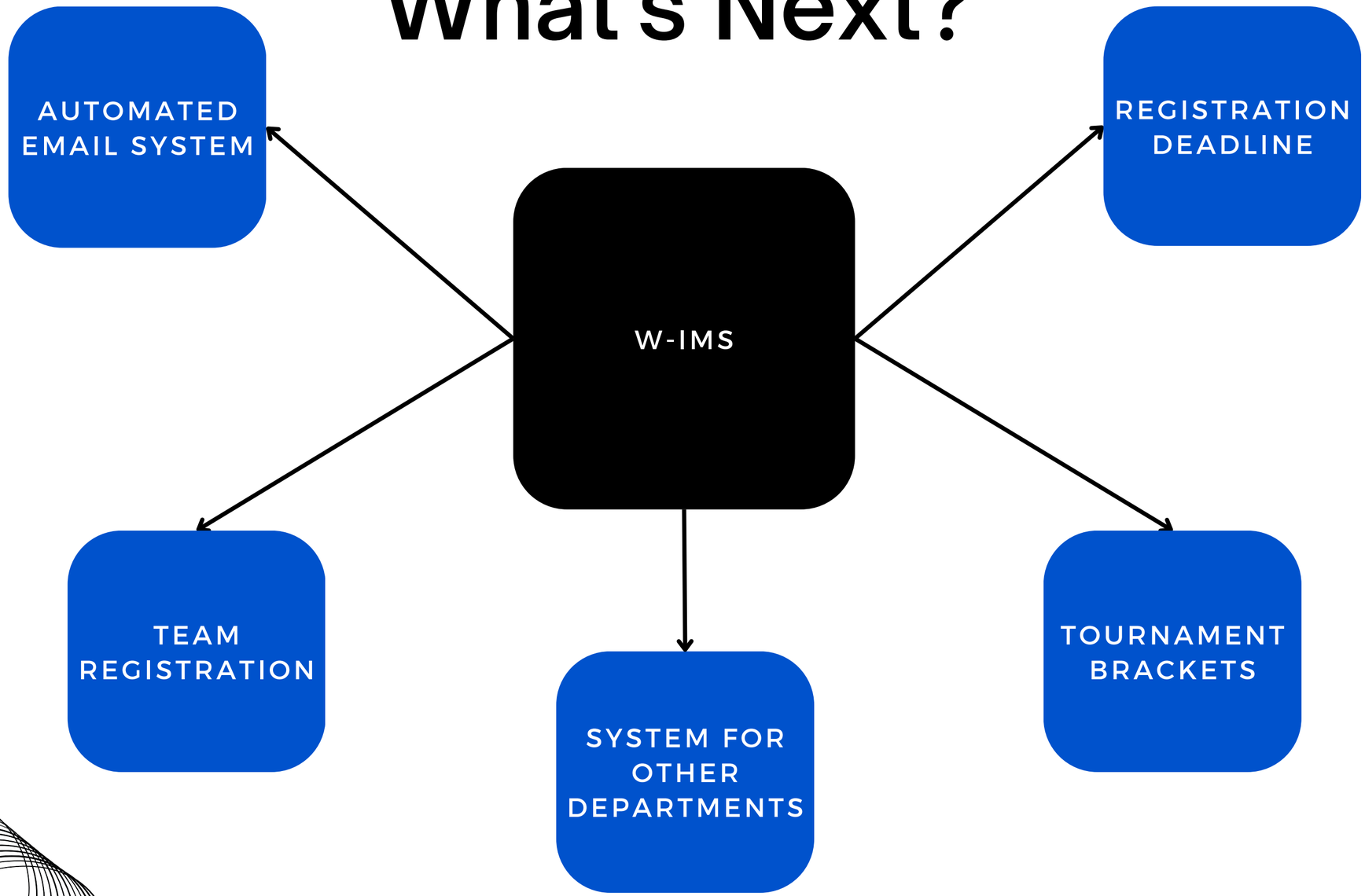
Scalable

```
const EventCreation = ({ user}) => {
  const [eventName, setEventName] = useState("");
  const [eventDate, setEventDate] = useState("");
  const [eventLocation, setEventLocation] = useState("");
  const [eventDescription, setEventDescription] = useState("");

  const eventsRef = collection(db, "events");

  const handleSubmit = async (e) => {
    e.preventDefault();
    setEventName("");
    setEventDate("");
    setEventLocation("");
    setEventDescription("");
    alert("Event Successfully Created!");

    const eventData = {
      eventName,
      eventDate,
      eventLocation,
      eventDescription,
    };

    try {
      await addDoc(eventsRef, eventData);
      // Handle success
    } catch (error) {
      // Handle error
    }
  };
```



- **Improved Search Engine Optimization**
- **Enhanced performance**
- **Lazy loading(Dynamic)**
- **Security is of the highest priority.**
- **Static site generation**

What's Next?


AUTOMATED EMAIL SYSTEM
REGISTRATION DEADLINE
W-IMS
TEAM REGISTRATION
SYSTEM FOR OTHER DEPARTMENTS
TOURNAMENT BRACKETS

Things that went well

- Explored options for project beyond class
- University likely to adopt system
- Small crunch time during SSRD
- Swapped tools when one wasn't working

Improvements going forward

- Meeting more frequently/regularly to update each other
- More frequent & clear communication
- Even distribution of Work

Overall Learning Outcome

- Software Development Methodologies - Agile Development
- Learned New technologies
- Do a lot of git commits
- Overall software development feel

Thank You

Questions?