

Turf Management System Case Study

File Tree

/turf-app

└─ db.json

└─ index.html

└─ package.json

└─ vite.config.js

└─ src/

└─ main.jsx

└─ App.jsx

└─ index.css

└─ utils/

| └─ auth.js

└─ services/

| └─ ApiService.js

└─ components/

| └─ AdminNavbar.jsx

| └─ UserNavbar.jsx

| └─ TurfCard.jsx

└─ pages/

└─ Welcome.jsx

└─ Login.jsx

└─ Register.jsx

└─ AdminDashboard.jsx

└─ UserDashboard.jsx

└─ Cart.jsx

└─ Checkout.jsx

└─ EditTurf.jsx

package.json

```
{  
  "name": "turf-app",  
  "private": true,  
  "version": "0.0.0",  
  "type": "module",  
  "scripts": {  
    "dev": "vite",  
    "build": "vite build",  
    "preview": "vite preview",  
    "server": "json-server --watch db.json --port 5000"  
  },  
  "dependencies": {  
    "axios": "^1.7.2",  
    "bootstrap": "^5.3.3",  
    "formik": "^2.4.5",  
    "react": "^18.2.0",  
    "react-dom": "^18.2.0",  
    "react-router-dom": "^6.26.1",  
    "yup": "^1.4.0"
```

```
},  
"devDependencies": {  
  "@vitejs/plugin-react": "^4.3.1",  
  "json-server": "^0.17.4",  
  "vite": "^5.4.2"  
}  
}
```

vite.config.js

```
import { defineConfig } from 'vite'  
import react from '@vitejs/plugin-react'  
  
export default defineConfig({  
  plugins: [react()],  
  server: { port: 5173 }  
})
```

index.html

```
<!doctype html>  
<html lang="en">  
  <head>  
    <meta charset="UTF-8" />  
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />  
    <title>Turf Management System</title>  
  </head>  
  <body>
```

```
<div id="root"></div>

<script type="module" src="/src/main.jsx"></script>

</body>

</html>
```

src/main.jsx

```
import React from 'react'

import ReactDOM from 'react-dom/client'

import { BrowserRouter } from 'react-router-dom'

import 'bootstrap/dist/css/bootstrap.min.css'

import './index.css'

import App from './App'

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <BrowserRouter>
      <App />
    </BrowserRouter>
  </React.StrictMode>
)
```

src/index.css

```
body { background: #f8f9fa; }

.navbar-brand { font-weight: 700; }

.card { border-radius: 1rem; }
```

src/utils/auth.js

```
const AUTH_KEY = 'authUser'
```

```
export function setAuth(user) {  
  localStorage.setItem(AUTH_KEY, JSON.stringify(user))  
}
```

```
export function getAuth() {  
  const raw = localStorage.getItem(AUTH_KEY)  
  return raw ? JSON.parse(raw) : null  
}
```

```
export function clearAuth() {  
  localStorage.removeItem(AUTH_KEY)  
}
```

```
export function isAdmin() {  
  const u = getAuth()  
  return !!u && u.role === 'admin'  
}
```

```
export function isUser() {  
  const u = getAuth()  
  return !!u && u.role === 'user'  
}
```

src/services/ApiService.js

```
import axios from 'axios'
```

```
const api = axios.create({ baseURL: 'http://localhost:5000' })
```

```
// USERS
```

```
export const getUserByCredentials = async (email, password) => {  
  const { data } = await api.get('/users', { params: { email, password } })  
  return data[0] || null  
}
```

```
export const registerUser = async (payload) => {  
  const { data } = await api.post('/users', payload)  
  return data  
}
```

```
// TURFS
```

```
export const getTurfs = async () => (await api.get('/turfs')).data  
export const getTurf = async (id) => (await api.get(`/turfs/${id}`)).data  
export const createTurf = async (payload) => (await api.post('/turfs', payload)).data  
export const updateTurf = async (id, payload) => (await api.put(`/turfs/${id}`,  
  payload)).data  
export const deleteTurf = async (id) => (await api.delete(`/turfs/${id}`)).data
```

```
// CART (per-user via userId)
```

```
export const getCartByUser = async (userId) => (await api.get('/cart', { params: { userId }  
  })).data  
export const addOrIncCartItem = async ({ userId, turfId, title, price }) => {
```

```

const { data: existing } = await api.get('/cart', { params: { userId, turfId } })
if (existing.length) {
  const item = existing[0]
  return (await api.patch(`/cart/${item.id}`, { quantity: (item.quantity || 1) + 1 })).data
}
return (await api.post('/cart', { userId, turfId, title, price, quantity: 1 })).data
}

export const updateCartQty = async (id, quantity) => (await api.patch(`/cart/${id}`, {
  quantity })).data

export const removeCartItem = async (id) => (await api.delete(`/cart/${id}`)).data

export const clearCartForUser = async (userId) => {
  const { data } = await api.get('/cart', { params: { userId } })
  await Promise.all(data.map(i => api.delete(`/cart/${i.id}`)))
}

// BOOKINGS

export const getAllBookings = async () => (await api.get('/bookings')).data

export const getBookingsByUser = async (userId) => (await api.get('/bookings', { params: {
  userId } })).data

export const createBooking = async (payload) => (await api.post('/bookings', payload)).data

```

src/components/AdminNavbar.jsx

```

import { Link, useNavigate } from 'react-router-dom'

import { clearAuth } from '../utils/auth'

export default function AdminNavbar() {
  const navigate = useNavigate()

```

```

const logout = () => { clearAuth(); navigate('/') }

return (
  <nav className="navbar navbar-expand navbar-dark bg-dark mb-4">
    <div className="container">
      <Link className="navbar-brand" to="/admin">Turf Admin</Link>
      <div className="navbar-nav">
        <Link className="nav-link" to="/admin">Home</Link>
        <Link className="nav-link" to="/admin">Manage Turfs</Link>
        <Link className="nav-link" to="/admin?tab=bookings">View Bookings</Link>
        <button className="btn btn-outline-light ms-3" onClick={logout}>Logout</button>
      </div>
    </div>
  </nav>
)
}

```

src/components/UserNavbar.jsx

```

import { Link, useNavigate } from 'react-router-dom'
import { clearAuth } from '../utils/auth'

export default function UserNavbar() {
  const navigate = useNavigate()
  const logout = () => { clearAuth(); navigate('/') }
  return (
    <nav className="navbar navbar-expand navbar-dark bg-primary mb-4">
      <div className="container">

```



```

<Link className="navbar-brand" to="/user">Turf</Link>

<div className="navbar-nav">

  <Link className="nav-link" to="/user">Browse Turfs</Link>

  <Link className="nav-link" to="/cart">Cart</Link>

  <Link className="nav-link" to="/checkout">Checkout</Link>

  <button className="btn btn-outline-light ms-3" onClick={logout}>Logout</button>

</div>

</div>

</nav>

)
}

```

src/components/TurfCard.jsx

```

export default function TurfCard({ turf, onAdd }) {

  return (

    <div className="col-md-4 mb-3">

      <div className="card h-100 shadow-sm">

        <div className="card-body">

          <h5 className="card-title">{turf.title}</h5>

          <p className="card-text">{turf.description}</p>

          <p className="card-text"><small className="text-muted">{turf.location}</small></p>

          <p className="fw-bold">₹ {turf.price}</p>

          {onAdd && (

            <button className="btn btn-primary" onClick={() => onAdd(turf)}>Add to
            Cart</button>

          )}

        </div>

      </div>

    </div>

  )
}

```

```
    </div>
  </div>
</div>
)
}
```

src/pages/Welcome.jsx

```
import { Link } from 'react-router-dom'

export default function Welcome() {
  return (
    <div className="container py-5 text-center">
      <h1 className="mb-3">Welcome to Turf Management System</h1>
      <p className="text-muted mb-4">Browse, book, and manage sports turfs with ease.</p>
      <div className="d-flex gap-3 justify-content-center">
        <Link to="/login" className="btn btn-primary btn-lg">Login</Link>
        <Link to="/register" className="btn btn-outline-primary btn-lg">Register</Link>
      </div>
    </div>
  )
}
```

src/pages/Register.jsx

```
import { Formik, Form, Field, ErrorMessage } from 'formik'
import * as Yup from 'yup'
```

```

import { registerUser } from '../services/ApiService'

import { Link, useNavigate } from 'react-router-dom'

const schema = Yup.object({
  name: Yup.string().required('Name is required'),
  email: Yup.string().email('Invalid email').required('Email is required'),
  password: Yup.string().min(6, 'Min 6 chars').required('Password is required')
})

export default function Register() {
  const navigate = useNavigate()

  return (
    <div className="container py-4" style={{maxWidth: 480}}>
      <h2 className="mb-3">Register</h2>
      <Formik
        initialValues={{ name: '', email: '', password: '' }}
        validationSchema={schema}
        onSubmit={async (values, { setSubmitting, resetForm }) => {
          try {
            await registerUser({ ...values, role: 'user' })
            resetForm()
            alert('Registration successful! Please login.')
            navigate('/login')
          } catch (e) {
            console.error(e)
          }
        }}
      >
    </div>
  )
}

```

```

    alert('Registration failed')
  } finally { setSubmitting(false) }
}
>
{{{ isSubmitting }} => (
  <Form className="card p-4 shadow-sm">
    <div className="mb-3">
      <label className="form-label">Name</label>
      <Field name="name" className="form-control" />
      <div className="text-danger small"><ErrorMessage name="name" /></div>
    </div>
    <div className="mb-3">
      <label className="form-label">Email</label>
      <Field name="email" type="email" className="form-control" />
      <div className="text-danger small"><ErrorMessage name="email" /></div>
    </div>
    <div className="mb-3">
      <label className="form-label">Password</label>
      <Field name="password" type="password" className="form-control" />
      <div className="text-danger small"><ErrorMessage name="password" /></div>
    </div>
    <button className="btn btn-primary" type="submit"
disabled={isSubmitting}>Create Account</button>
    <Link to="/login" className="btn btn-link">Have an account? Login</Link>
  </Form>
)}

```

```
    </Formik>
  </div>
)
}
```

src/pages/Login.jsx

```
import { Formik, Form, Field, ErrorMessage } from 'formik'
import * as Yup from 'yup'
import { getUserByCredentials } from '../services/ApiService'
import { setAuth } from '../utils/auth'
import { useNavigate, Link } from 'react-router-dom'

const schema = Yup.object({
  email: Yup.string().email('Invalid email').required('Required'),
  password: Yup.string().required('Required')
})

export default function Login() {
  const navigate = useNavigate()

  return (
    <div className="container py-4" style={{maxWidth: 480}}>
      <h2 className="mb-3">Login</h2>
      <Formik
        initialValues={{ email: '', password: '' }}
        validationSchema={schema}

```

```
onSubmit={async (values, { setSubmitting }) => {
```

```
  try {
```

```
    const user = await getUserByCredentials(values.email, values.password)
```

```
    if (!user) { alert('Invalid credentials'); return }
```

```
    setAuth(user)
```

```
    if (user.role === 'admin') navigate('/admin')
```

```
    else navigate('/user')
```

```
  } catch (e) {
```

```
    console.error(e)
```

```
    alert('Login failed')
```

```
  } finally { setSubmitting(false) }
```

```
}}
```

```
>
```

```
{{{ isSubmitting }} => (
```

```
  <Form className="card p-4 shadow-sm">
```

```
    <div className="mb-3">
```

```
      <label className="form-label">Email</label>
```

```
      <Field name="email" type="email" className="form-control" />
```

```
      <div className="text-danger small"><ErrorMessage name="email" /></div>
```

```
    </div>
```

```
    <div className="mb-3">
```

```
      <label className="form-label">Password</label>
```

```
      <Field name="password" type="password" className="form-control" />
```

```
      <div className="text-danger small"><ErrorMessage name="password" /></div>
```

```
    </div>
```

```
        <button className="btn btn-primary" type="submit"
disabled={isSubmitting}>Login</button>

        <Link to="/register" className="btn btn-link">Create an account</Link>

    </Form>

  })

</Formik>

</div>

)

}
```

src/pages/AdminDashboard.jsx

```
import { useEffect, useMemo, useState } from 'react'

import { Link, useLocation, useNavigate } from 'react-router-dom'

import AdminNavbar from '../components/AdminNavbar'

import { createTurf, deleteTurf, getAllBookings, getTurfs } from '../services/ApiService'

import { Formik, Form, Field, ErrorMessage } from 'formik'

import * as Yup from 'yup'

import { getAuth } from '../utils/auth'

const turfSchema = Yup.object({

  title: Yup.string().required('Required'),

  location: Yup.string().required('Required'),

  price: Yup.number().positive().required('Required'),

  description: Yup.string().required('Required')

})
```

```
export default function AdminDashboard() {  
  const [turfs, setTurfs] = useState([])  
  const [bookings, setBookings] = useState([])  
  const location = useLocation()  
  const tab = new URLSearchParams(location.search).get('tab') || 'turfs'  
  const navigate = useNavigate()  
  
  useEffect(() => {  
    const me = getAuth()  
    if (!me || me.role !== 'admin') navigate('/login')  
  }, [])  
  
  const load = async () => {  
    setTurfs(await getTurfs())  
  }  
  const loadBookings = async () => {  
    setBookings(await getAllBookings())  
  }  
  
  useEffect(() => { load() }, [])  
  useEffect(() => { if (tab === 'bookings') loadBookings() }, [tab])  
  
  return (  
    <>  
    <AdminNavbar />  
    <div className="container">
```



```

<ul className="nav nav-tabs mb-3">
  <li className="nav-item"><Link className={` nav-link ${tab==='turfs'? 'active':''}` }
to="/admin?tab=turfs">Manage Turfs</Link></li>

  <li className="nav-item"><Link className={` nav-link
${tab==='bookings'? 'active':''}` } to="/admin?tab=bookings">View Bookings</Link></li>

</ul>

```

```

{tab === 'turfs' && (
  <div className="row">
    <div className="col-md-7">
      <h4 className="mb-3">All Turfs</h4>
      <div className="list-group mb-4">
        {turfs.map(t => (
          <div key={t.id} className="list-group-item d-flex justify-content-between align-
items-center">
            <div>
              <div className="fw-bold">{t.title}</div>
              <small className="text-muted">{t.location} • ₹{t.price}</small>
            </div>
            <div>
              <Link className="btn btn-sm btn-outline-secondary me-2" to={` /edit-
turf/${t.id}` }>Edit</Link>
              <button className="btn btn-sm btn-outline-danger" onClick={async()=>{
if(confirm('Delete turf?')){ await deleteTurf(t.id); load() } }}>Delete</button>
            </div>
          </div>
        ))}
      </div>
    </div>
  </div>
)
}

```

```

    {!turfs.length && <div className="text-muted">No turfs yet.</div>}
  </div>
</div>

<div className="col-md-5">
  <h4 className="mb-3">Add Turf</h4>
  <Formik
    initialValues={{ title:"", location:"", price:"", description:"" }}
    validationSchema={turfSchema}
    onSubmit={async (values, { resetForm }) => {
      await createTurf({ ...values, price: Number(values.price) })
      resetForm();
      load();
    }}
  >
    {() => (
      <Form className="card p-3 shadow-sm">
        <div className="mb-2">
          <label className="form-label">Title</label>
          <Field name="title" className="form-control" />
          <div className="text-danger small"><ErrorMessage name="title"/></div>
        </div>
        <div className="mb-2">
          <label className="form-label">Location</label>
          <Field name="location" className="form-control" />
          <div className="text-danger small"><ErrorMessage name="location"/></div>

```

```

    </div>

    <div className="mb-2">
      <label className="form-label">Price</label>

      <Field name="price" type="number" className="form-control" />

      <div className="text-danger small"><ErrorMessage name="price"/></div>
    </div>

    <div className="mb-2">
      <label className="form-label">Description</label>

      <Field as="textarea" rows="3" name="description" className="form-control" />

      <div className="text-danger small"><ErrorMessage
name="description"/></div>
    </div>

    <button className="btn btn-success">Add Turf</button>

  </Form>

  )}
</Formik>
</div>
</div>
)}

```

```

{tab === 'bookings' && (
  <div className="card p-3 shadow-sm">
    <h4 className="mb-3">All Bookings</h4>
    <div className="table-responsive">
      <table className="table table-striped">
        <thead>

```

```

<tr>

  <th>#</th>

  <th>User ID</th>

  <th>Turf ID</th>

  <th>Title</th>

  <th>Qty</th>

  <th>Amount</th>

  <th>Date</th>

</tr>

</thead>

<tbody>

  {bookings.map((b, idx) => (

    <tr key={b.id}>

      <td>{idx+1}</td>

      <td>{b.userId}</td>

      <td>{b.turfId}</td>

      <td>{b.title}</td>

      <td>{b.quantity}</td>

      <td>₹{b.amount}</td>

      <td>{new Date(b.createdAt).toLocaleString()}</td>

    </tr>

  ))}

  {!bookings.length && (

    <tr><td colspan="7" className="text-muted">No bookings yet.</td></tr>

  )}

</tbody>

```

```
        </table>
      </div>
    </div>
  )}
</div>
</>
)
}
```

src/pages/UserDashboard.jsx

```
import { useEffect, useState } from 'react'
import UserNavbar from '../components/UserNavbar'
import TurfCard from '../components/TurfCard'
import { addOrIncCartItem, getTurfs } from '../services/ApiService'
import { getAuth } from '../utils/auth'
import { useNavigate } from 'react-router-dom'

export default function UserDashboard() {
  const [turfs, setTurfs] = useState([])
  const navigate = useNavigate()

  useEffect(() => {
    const me = getAuth()
    if (!me) { navigate('/login'); return }
    setTurfs([])
    getTurfs().then(setTurfs)
  })
}
```

```
}, [])
```

```
const handleAdd = async (turf) => {  
  const me = getAuth(); if (!me) return navigate('/login')  
  await addOrIncCartItem({ userId: me.id, turfId: turf.id, title: turf.title, price: turf.price })  
  alert('Added to cart')  
}
```

```
return (  
  <>  
    <UserNavbar />  
    <div className="container">  
      <h3 className="mb-3">Browse Turfs</h3>  
      <div className="row">  
        {turfs.map(t => (  
          <TurfCard key={t.id} turf={t} onAdd={handleAdd} />  
        ))}  
        {!turfs.length && <div className="text-muted">No turfs found.</div>}  
      </div>  
    </div>  
  </>  
)  
}
```

src/pages/Cart.jsx

```
import { useEffect, useState } from 'react'
```

```
import UserNavbar from '../components/UserNavbar'

import { getCartByUser, removeCartItem, updateCartQty } from '../services/ApiService'

import { getAuth } from '../utils/auth'

import { useNavigate } from 'react-router-dom'
```

```
export default function Cart() {

  const [items, setItems] = useState([])

  const navigate = useNavigate()

  const load = async () => {

    const me = getAuth()

    if (!me) return navigate('/login')

    setItems(await getCartByUser(me.id))

  }
```

```
  useEffect(() => { load() }, [])
```

```
  const total = items.reduce((sum, i) => sum + (i.price * (i.quantity || 1)), 0)
```

```
  return (

    <>

      <UserNavbar />

      <div className
```