**Project code**

#### **src/components/BillSummary.tsx**

import React from 'react';

import { Card, CardContent, CardDescription, CardFooter, CardHeader, CardTitle } from '@/components/ui/card';

import { Button } from '@/components/ui/button';

import { Table, TableBody, TableCaption, TableCell, TableHead, TableHeader, TableRow } from '@/components/ui/table';

import { Calendar, Download } from 'lucide-react';

interface BillItem { date: string; breakfast: boolean; lunch: boolean; dinner: boolean;

totalAmount: number;

}

interface BillSummaryProps { month: string;

year: number; billItems: BillItem[]; totalAmount: number; dueDate: string;

}

const BillSummary: React.FC<BillSummaryProps> = ({ month,

year, billItems, totalAmount, dueDate

}) => {

const mealRates = { breakfast: 50,

lunch: 100,

dinner: 100

};

const downloadBill = () => {

// In a real application, this would generate a PDF or CSV console.log('Downloading bill...');

};

return (

<Card className="animate-fade-in">

<CardHeader>

<CardTitle className="text-mess-blue">Monthly Bill Summary</CardTitle>

<CardDescription className="flex items-center gap-1">

<Calendar className="h-4 w-4" /> {month} {year}

</CardDescription>

</CardHeader>

<CardContent className="space-y-4">

<div className="bg-mess-lightGray rounded-md p-4 flex flex-col sm:flex-row justify- between gap-4">

<div>

<p className="text-sm text-muted-foreground">Total Amount</p>

<p className="text-2xl font-bold text-mess-blue">₹{totalAmount.toFixed(2)}</p>

</div>

<div>

<p className="text-sm text-muted-foreground">Due Date</p>

<p className="text-md font-semibold text-mess-darkGray">{dueDate}</p>

</div>

<div>

<p className="text-sm text-muted-foreground">Status</p>

<span className="inline-flex items-center rounded-full bg-green-100 px-3 py-0.5 text-sm font-medium text-green-800">

Unpaid

</span>

</div>

</div>

<div className="overflow-x-auto">

<Table>

<TableCaption>Bill breakdown for {month} {year}</TableCaption>

<TableHeader>

<TableRow>

<TableHead>Date</TableHead>

<TableHead>Breakfast (₹{mealRates.breakfast})</TableHead>

<TableHead>Lunch (₹{mealRates.lunch})</TableHead>

<TableHead>Dinner (₹{mealRates.dinner})</TableHead>

<TableHead className="text-right">Amount</TableHead>

</TableRow>

</TableHeader>

<TableBody>

{billItems.map((item) => (

<TableRow key={item.date}>

<TableCell className="font-medium">{item.date}</TableCell>

<TableCell>{item.breakfast ? '✓' : '-'}</TableCell>

<TableCell>{item.lunch ? '✓' : '-'}</TableCell>

<TableCell>{item.dinner ? '✓' : '-'}</TableCell>

<TableCell className="text-right">₹{item.totalAmount.toFixed(2)}</TableCell>

</TableRow>

))}

</TableBody>

</Table>

</div>

</CardContent>

<CardFooter>

<Button onClick={downloadBill} variant="outline"

className="w-full flex items-center gap-2"

>

<Download className="h-4 w-4" /> Download Bill

</Button>

</CardFooter>

</Card>

);

};

export default BillSummary;

#### **src/components/BookingList.tsx**

import React, { useState } from 'react';

import { Card, CardContent, CardDescription, CardHeader, CardTitle } from '@/components/ui/card';

import { Button } from '@/components/ui/button'; import { Input } from '@/components/ui/input'; import { Calendar } from '@/components/ui/calendar';

import { Popover, PopoverContent, PopoverTrigger } from '@/components/ui/popover'; import { Table, TableBody, TableCell, TableHead, TableHeader, TableRow } from '@/components/ui/table';

import { Badge } from '@/components/ui/badge'; import { format } from 'date-fns';

import { Calendar as CalendarIcon, Check, X, Download } from 'lucide-react'; import { Select, SelectContent, SelectItem, SelectTrigger, SelectValue } from '@/components/ui/select';

interface BookingData { id: string;

studentId: string; studentName: string; date: Date; breakfast: boolean; lunch: boolean; dinner: boolean;

}

const BookingList: React.FC = () => {

const [date, setDate] = useState<Date>(new Date()); const [search, setSearch] = useState<string>('');

const [mealFilter, setMealFilter] = useState<string>('all');

// Mock data

const bookings: BookingData[] = [

{

id: '1',

studentId: 'S12345', studentName: 'Rahul Sharma', date: new Date(),

breakfast: true, lunch: true, dinner: false,

},

{

id: '2',

studentId: 'S12346', studentName: 'Priya Singh', date: new Date(), breakfast: false,

lunch: true, dinner: true,

},

{

id: '3',

studentId: 'S12347', studentName: 'Amit Kumar', date: new Date(),

breakfast: true, lunch: false, dinner: true,

},

{

id: '4',

studentId: 'S12348', studentName: 'Neha Gupta', date: new Date(),

breakfast: true, lunch: true, dinner: true,

},

];

// Filter bookings based on search and meal filter const filteredBookings = bookings.filter(booking => { const matchesSearch =

booking.studentName.toLowerCase().includes(search.toLowerCase()) || booking.studentId.toLowerCase().includes(search.toLowerCase());

let matchesMeal = true;

if (mealFilter === 'breakfast') matchesMeal = booking.breakfast; if (mealFilter === 'lunch') matchesMeal = booking.lunch;

if (mealFilter === 'dinner') matchesMeal = booking.dinner;

return matchesSearch && matchesMeal;

});

const downloadReport = () => {

// In a real app, this would generate a CSV or PDF console.log('Downloading bookings report...');

};

// Count totals

const totalBreakfast = filteredBookings.filter(b => b.breakfast).length; const totalLunch = filteredBookings.filter(b => b.lunch).length;

const totalDinner = filteredBookings.filter(b => b.dinner).length;

return (

<div className="space-y-6 animate-fade-in">

<Card>

<CardHeader>

<CardTitle className="text-mess-blue">Meal Bookings</CardTitle>

<CardDescription>

View and manage student meal bookings

</CardDescription>

</CardHeader>

<CardContent className="space-y-4">

<div className="flex flex-col sm:flex-row gap-4">

<div className="flex-1">

<Popover>

<PopoverTrigger asChild>

<Button variant="outline"

className="w-full justify-start text-left font-normal"

>

<CalendarIcon className="mr-2 h-4 w-4" />

{format(date, 'PPP')}

</Button>

</PopoverTrigger>

<PopoverContent className="w-auto p-0">

<Calendar mode="single" selected={date}

onSelect={(newDate) => newDate && setDate(newDate)} initialFocus

/>

</PopoverContent>

</Popover>

</div>

<div className="flex-1">

<Input

placeholder="Search by name or ID" value={search}

onChange={(e) => setSearch(e.target.value)} className="w-full"

/>

</div>

<div className="w-40">

<Select value={mealFilter} onValueChange={setMealFilter}>

<SelectTrigger>

<SelectValue placeholder="Filter by meal" />

</SelectTrigger>

<SelectContent>

<SelectItem value="all">All Meals</SelectItem>

<SelectItem value="breakfast">Breakfast</SelectItem>

<SelectItem value="lunch">Lunch</SelectItem>

<SelectItem value="dinner">Dinner</SelectItem>

</SelectContent>

</Select>

</div>

<Button variant="outline"

className="flex items-center gap-2" onClick={downloadReport}

>

<Download className="h-4 w-4" /> Export

</Button>

</div>

<div className="flex flex-wrap gap-3 pb-2">

<Badge variant="outline" className="bg-mess-lightGray"> Breakfast: {totalBreakfast}

</Badge>

<Badge variant="outline" className="bg-mess-lightGray"> Lunch: {totalLunch}

</Badge>

<Badge variant="outline" className="bg-mess-lightGray"> Dinner: {totalDinner}

</Badge>

<Badge variant="outline" className="bg-mess-lightGray"> Total Students: {filteredBookings.length}

</Badge>

</div>

<div className="overflow-x-auto">

<Table>

<TableHeader>

<TableRow>

<TableHead>Student ID</TableHead>

<TableHead>Name</TableHead>

<TableHead className="text-center">Breakfast</TableHead>

<TableHead className="text-center">Lunch</TableHead>

<TableHead className="text-center">Dinner</TableHead>

</TableRow>

</TableHeader>

<TableBody>

{filteredBookings.length === 0 ? (

<TableRow>

<TableCell colSpan={5} className="text-center text-muted-foreground"> No bookings found

</TableCell>

</TableRow>

) : (

filteredBookings.map((booking) => (

<TableRow key={booking.id}>

<TableCell>{booking.studentId}</TableCell>

<TableCell>{booking.studentName}</TableCell>

<TableCell className="text-center">

{booking.breakfast ? (

<Check className="h-4 w-4 text-green-600 mx-auto" />

) : (

<X className="h-4 w-4 text-red-600 mx-auto" />

)}

</TableCell>

<TableCell className="text-center">

{booking.lunch ? (

<Check className="h-4 w-4 text-green-600 mx-auto" />

) : (

<X className="h-4 w-4 text-red-600 mx-auto" />

)}

</TableCell>

<TableCell className="text-center">

{booking.dinner ? (

<Check className="h-4 w-4 text-green-600 mx-auto" />

) : (

<X className="h-4 w-4 text-red-600 mx-auto" />

)}

</TableCell>

</TableRow>

))

)}

</TableBody>

</Table>

</div>

</CardContent>

</Card>

</div>

);

};

export default BookingList;