

```

'use client'

import { useEffect, useState } from 'react'
import { Building, Clock, Phone, Mail, Save, Calendar, Plug, Loader2 } from 'lucide-react'
import { Button } from '@components/ui/button'
import { Card, CardContent, CardDescription, CardHeader, CardTitle } from '@components/ui/card'
import { Input } from '@components/ui/input'
import { Textarea } from '@components/ui/textarea'
import { Badge } from '@components/ui/badge'
import { Switch } from '@components/ui/switch'
import { Label } from '@components/ui/label'
import { useLayout } from '@components/ui/layout-context'
import { Tabs, TabsContent, TabsList, TabsTrigger } from '@components/ui/tabs'
import { useBusinessSettings } from '@hooks/useBusinessSettings'

interface BusinessSettings {
  name: string
  description: string
  phone: string
  email: string
  address: string
  city: string
  country: string
  timezone: string
  website: string
  businessHours: {
    [ key: string ]: { open: string; close: string; isOpen: boolean }
  }
  notifications: {
    emailBookings: boolean
    smsReminders: boolean
    callAlerts: boolean
  }
  integrations: {
    googleCalendar: { enabled: boolean; connected: boolean }
  }
}

```

```
whatsapp: { enabled: boolean; connected: boolean }
vapi: { enabled: boolean; connected: boolean }
}
}

// Default settings to use while loading
const defaultSettings: BusinessSettings = {
  name: '',
  description: '',
  phone: '',
  email: '',
  address: '',
  city: '',
  country: '',
  timezone: 'America/Los_Angeles',
  website: '',
  businessHours: {
    monday: { open: '09:00', close: '18:00', isOpen: true },
    tuesday: { open: '09:00', close: '18:00', isOpen: true },
    wednesday: { open: '09:00', close: '18:00', isOpen: true },
    thursday: { open: '09:00', close: '18:00', isOpen: true },
    friday: { open: '09:00', close: '18:00', isOpen: true },
    saturday: { open: '10:00', close: '16:00', isOpen: true },
    sunday: { open: '10:00', close: '16:00', isOpen: false }
  },
  notifications: {
    emailBookings: true,
    smsReminders: true,
    callAlerts: true
  },
  integrations: {
    googleCalendar: { enabled: false, connected: false },
    whatsapp: { enabled: false, connected: false },
    vapi: { enabled: false, connected: false }
  }
}
```

```

export default function SettingsPage() {
  const { setTitle, setSubtitle } = useLayout()

  // Use the business settings hook
  const {
    settings: fetchedSettings,
    isLoading,
    error,
    updateSettings,
    isUpdating
  } = useBusinessSettings()

  // Local state for form values
  const [ settings, setSettings ] = useState(defaultSettings)
  const [ hasChanges, setHasChanges ] = useState(false)

  // Initialize settings when data is loaded
  useEffect(() => {
    if (fetchedSettings && !isLoading) {
      setSettings(fetchedSettings)
    }
  }, [fetchedSettings, isLoading])

  useEffect(() => {
    setTitle('Settings');
    setSubtitle('Manage your business configuration and preferences');
  }, []);

  const handleSettingChange = (path: string, value: any) => {
    setSettings(prev => {
      const keys = path.split('.')
      const newSettings = JSON.parse(JSON.stringify(prev))
      let current = newSettings

```

```
    for (let i = 0; i < keys.length - 1; i++) {  
      current = current[ keys[ i ] ]  
    }  
    current[ keys[ keys.length - 1 ] ] = value  
  
    return newSettings  
  })  
  setHasChanges(true)
```

```
}  
  
const handleSave = async () => {  
  const success = await updateSettings(settings)  
  if (success) {  
    setHasChanges(false)  
  }  
}  
  
const daysOfWeek = [  
  { key: 'monday', label: 'Monday' },  
  { key: 'tuesday', label: 'Tuesday' },  
  { key: 'wednesday', label: 'Wednesday' },  
  { key: 'thursday', label: 'Thursday' },  
  { key: 'friday', label: 'Friday' },  
  { key: 'saturday', label: 'Saturday' },  
  { key: 'sunday', label: 'Sunday' }  
]  
  
const timezones = [  
  'America/New_York',  
  'America/Chicago',  
  'America/Denver',  
  'America/Los_Angeles',  
  'Europe/London',  
  'Europe/Paris',  
  'Asia/Tokyo',
```

'Australia/Sydney'

]

return (

Business Info

Business Hours

Notifications

Integrations

{hasChanges && (



```
{isUpdating ? (
```

```
) : (
```

```
)}
```

```
{isUpdating ? 'Saving...' : 'Save Changes'}
```

```
)}
```

```

    { /* Business Information */ }
    <TabsContent value="business-information">
      <Card className='border pt-0 overflow-hidden h-full shadow-
none'>
        <CardHeader className='bg-neutral-100 border-b border-
neutral-200 py-4 gap-0'>
          <CardTitle className="text-gray-900 text-sm font-
semibold">
            Business Information
          </CardTitle>
          <CardDescription className="text-gray-500 text-xs">Basic
information about your business</CardDescription>
        </CardHeader>
        <CardContent className="space-y-4">
          <div className="grid grid-cols-1 md:grid-cols-2 gap-4">
            <div>
              <Label htmlFor="businessName">Business Name</Label>
              <Input
                id="businessName"
                value={settings.name}
                onChange={(e) => handleSettingChange('name',
e.target.value)}
                className="mt-1"
              />
            </div>
            <div>
              <Label htmlFor="website">Website</Label>
              <Input
                id="website"
                value={settings.website}
                onChange={(e) => handleSettingChange('website',
e.target.value)}
                className="mt-1"
              />
            </div>
          </div>
        </CardContent>
      </Card>
    </TabsContent>
  </div>

```

```

    </div>
    <div>
      <Label htmlFor="description">Description</Label>
      <Textarea
        id="description"
        value={settings.description}
        onChange={(e) => handleSettingChange('description',
e.target.value)}
        rows={3}
        className="mt-1"
      />
    </div>
    <div className="grid grid-cols-1 md:grid-cols-2 gap-4">
      <div>
        <Label htmlFor="phone">Phone Number</Label>
        <Input
          id="phone"
          value={settings.phone}
          onChange={(e) => handleSettingChange('phone',
e.target.value)}
          className="mt-1"
        />
      </div>
      <div>
        <Label htmlFor="email">Email Address</Label>
        <Input
          id="email"
          type="email"
          value={settings.email}
          onChange={(e) => handleSettingChange('email',
e.target.value)}
          className="mt-1"
        />
      </div>
    </div>
  </div>

```



```

<div className="grid grid-cols-1 md:grid-cols-3 gap-4">
  <div>
    <Label htmlFor="address">Address</Label>
    <Input
      id="address"
      value={settings.address}
      onChange={(e) => handleSettingChange('address',
e.target.value)}
      className="mt-1"
    />
  </div>
  <div>
    <Label htmlFor="city">City</Label>
    <Input
      id="city"
      value={settings.city}
      onChange={(e) => handleSettingChange('city',
e.target.value)}
      className="mt-1"
    />
  </div>
  <div>
    <Label htmlFor="timezone">Timezone</Label>
    <select
      id="timezone"
      value={settings.timezone}
      onChange={(e) => handleSettingChange('timezone',
e.target.value)}
      className="w-full mt-1 px-3 py-2 border border-
gray-200 rounded-md focus:outline-none focus:ring-2 focus:ring-
primary"
    >
      {timezones.map(tz => (
        <option key={tz} value={tz}>{tz}</option>
      ))}
    </select>
  </div>
</div>

```

```

        </select>
      </div>
    </div>
  </CardContent>
</Card>
</TabsContent>

{/* Business Hours */}
<TabsContent value="business-hours">
  <Card className='border pt-0 overflow-hidden h-full shadow-
none'>
    <CardHeader className='bg-neutral-100 border-b border-
neutral-200 py-4 gap-0'>
      <CardTitle className="text-gray-900 text-sm font-
semibold">
        Business Hours
      </CardTitle>
      <CardDescription className="text-gray-500 text-xs">Set
your operating hours for each day of the week</CardDescription>
    </CardHeader>
    <CardContent>
      <div className="space-y-4">
        {daysOfWeek.map(day => (
          <div key={day.key} className="flex items-center
gap-4">
            <div className="w-24">
              <Label className="text-sm font-
medium">{day.label}</Label>
            </div>
            <div className="flex items-center gap-2">
              <Switch
                checked={settings.businessHours[
day.key ].isOpen}
                onCheckedChange={(checked) =>
                  handleSettingChange(`businessHours.$

```

```

{day.key}.isOpen`, checked)
    }
  />
  <Label className="text-sm">Open</Label>
</div>
{settings.businessHours[ day.key ].isOpen && (
  <>
    <div>
      <Input
        type="time"
        value={settings.businessHours[
day.key ].open}
        onChange={(e) =>
          handleSettingChange(`businessHours.$
{day.key}.open`, e.target.value)
        }
        className="w-32"
      />
    </div>
    <span className="text-gray-400">to</span>
    <div>
      <Input
        type="time"
        value={settings.businessHours[
day.key ].close}
        onChange={(e) =>
          handleSettingChange(`businessHours.$
{day.key}.close`, e.target.value)
        }
        className="w-32"
      />
    </div>
  </>
)}
</div>

```

```

        )))
    </div>
</CardContent>
</Card>
</TabsContent>

{/* Notifications */}
<TabsContent value="notifications">
    <Card className='border pt-0 overflow-hidden h-full shadow-
none'>
        <CardHeader className='bg-neutral-100 border-b border-
neutral-200 py-4 gap-0'>
            <CardTitle className="text-gray-900 text-sm font-
semibold">
                Notifications
            </CardTitle>
            <CardDescription className="text-gray-500 text-
xs">Configure how you want to receive notifications</
CardDescription>
        </CardHeader>
        <CardContent className="space-y-4">
            <div className="flex items-center justify-between">
                <div>
                    <Label className="font-medium">Email Booking
Confirmations</Label>
                    <p className="text-sm text-gray-500">Receive email
when new bookings are made</p>
                </div>
                <Switch
                    checked={settings.notifications.emailBookings}
                    onCheckedChange={(checked) =>
                        handleSettingChange('notifications.emailBookings',
checked)
                    }
                />
            </div>
        </CardContent>
    </Card>
</TabsContent>

```

```

    </div>
    <div className="flex items-center justify-between">
      <div>
        <Label className="font-medium">SMS Reminders</Label>
        <p className="text-sm text-gray-500">Send SMS
reminders to customers</p>
      </div>
      <Switch
        checked={settings.notifications.smsReminders}
        onCheckedChange={(checked) =>
          handleSettingChange('notifications.smsReminders',
checked)
        }
      />
    </div>
    <div className="flex items-center justify-between">
      <div>
        <Label className="font-medium">Call Alerts</Label>
        <p className="text-sm text-gray-500">Get notified
when calls are received</p>
      </div>
      <Switch
        checked={settings.notifications.callAlerts}
        onCheckedChange={(checked) =>
          handleSettingChange('notifications.callAlerts',
checked)
        }
      />
    </div>
  </CardContent>
</Card>
</TabsContent>

  {/* Integrations */}
  <TabsContent value="integrations">

```

```

    <Card className='border pt-0 overflow-hidden h-full shadow-
none'>
      <CardHeader className='bg-neutral-100 border-b border-
neutral-200 py-4 gap-0'>
        <CardTitle className="text-gray-900 text-sm font-
semibold">
          Integrations
        </CardTitle>
        <CardDescription className="text-gray-500 text-
xs">Manage third-party service connections</CardDescription>
      </CardHeader>
      <CardContent className="space-y-4">
        <div className="flex items-center justify-between p-4
border border-gray-200 rounded-lg">
          <div className="flex items-center gap-3">
            <div
className="w-10 h-10 bg-blue-100 rounded-lg flex items-center
justify-center">
              <Calendar className="w-5 h-5 text-blue-600" />
            </div>
            <div>
              <Label className="font-medium">Google Calendar</
Label>
              <p className="text-sm text-gray-500">Sync bookings
with Google Calendar</p>
            </div>
          </div>
          <div className="flex items-center gap-2">
            <Badge
className={settings.integrations.googleCalendar.connected
? 'bg-green-100 text-green-800'
: 'bg-red-100 text-red-800'
}
            >

```

```

        {settings.integrations.googleCalendar.connected ?
'Connected' : 'Disconnected'}
      </Badge>
      <Switch

checked={settings.integrations.googleCalendar.enabled}
      onChangeChange={(checked) =>

handleSettingChange('integrations.googleCalendar.enabled', checked)
        }
      />
    </div>
  </div>

  <div className="flex items-center justify-between p-4
border border-gray-200 rounded-lg">
    <div className="flex items-center gap-3">
      <div className="w-10 h-10 bg-green-100 rounded-lg
flex items-center justify-center">
        <Phone className="w-5 h-5 text-green-600" />
      </div>
      <div>
        <Label className="font-medium">WhatsApp Business</
Label>
        <p className="text-sm text-gray-500">Send booking
confirmations via WhatsApp</p>
      </div>
    </div>
    <div className="flex items-center gap-2">
      <Badge
        className={settings.integrations.whatsapp.connected
          ? 'bg-green-100 text-green-800'
          : 'bg-red-100 text-red-800'
        }
      >

```

```

        {settings.integrations.whatsapp.connected ?
'Connected' : 'Disconnected'}
      </Badge>
      <Switch
        checked={settings.integrations.whatsapp.enabled}
        onChange={(checked) =>
handleSettingChange('integrations.whatsapp.enabled', checked)
        }
      />
    </div>
  </div>

  <div className="flex items-center justify-between p-4
border border-gray-200 rounded-lg">
    <div className="flex items-center gap-3">
      <div className="w-10 h-10 bg-purple-100 rounded-lg
flex items-center justify-center">
        <Phone className="w-5 h-5 text-purple-600" />
      </div>
      <div>
        <Label className="font-medium">Vapi AI</Label>
        <p className="text-sm text-gray-500">Voice AI for
automated phone calls</p>
      </div>
    </div>
    <div className="flex items-center gap-2">
      <Badge
        className={settings.integrations.vapi.connected
? 'bg-green-100 text-green-800'
: 'bg-red-100 text-red-800'
        }
      >
        {settings.integrations.vapi.connected ?
'Connected' : 'Disconnected'}

```



```

        </Badge>
        <Switch
          checked={settings.integrations.vapi.enabled}
          onCheckedChange={({checked) =>
            handleSettingChange('integrations.vapi.enabled',
checked)
          }
        />
      </div>
    </div>
  </CardContent>
</Card>
</TabsContent>
</Tabs>
</div>

{/* Loading state */}
{isLoading && (
  <div className="fixed inset-0 bg-white/60 flex items-center
justify-center z-50">
    <div
className="flex items-center gap-2 p-4 bg-white rounded-lg shadow-
lg">
      <Loader2 className="w-5 h-5 animate-spin text-primary" />
      <p className="text-sm font-medium">Loading settings...</p>
    </div>
  </div>
)}

{/* Error state */}
{error && (
  <Card className="mt-6 border-red-200 bg-red-50">
    <CardContent className="pt-6">
      <p className="text-red-700">Error loading settings. Please
try again later.</p>

```

```
    </CardContent>  
  </Card>  
)}
```

```
)  
}
```