

Voici toutes les pages avancées complètes pour la plateforme AgroDeep, organisées selon la structure du projet :

## 📁 \*\*PAGES D'AUTHENTIFICATION\*\*

### \*\*`src/app/(auth)/login/page.tsx`\*\*

``tsx

'use client';

```
import { useState } from 'react';
import { useRouter } from 'next/navigation';
import { zodResolver } from '@hookform/resolvers/zod';
import { useForm } from 'react-hook-form';
import Link from 'next/link';
import { Button } from '@components/ui/button';
import { Input } from '@components/ui/input';
import { Card, CardContent, CardDescription, CardFooter, CardHeader, CardTitle } from '@components/ui/card';
import { Label } from '@components/ui/label';
import { Checkbox } from '@components/ui/checkbox';
import { Separator } from '@components/ui/separator';
import { loginSchema } from '@types/schema/users.schema';
import { useAuthStore } from '@stores/useAuthStore';
import { logger } from '@services/monitoring/logger';
import { Eye, EyeOff, Lock, Mail } from 'lucide-react';
```

```
export default function LoginPage() {
  const [showPassword, setShowPassword] = useState(false);
  const [isLoading, setIsLoading] = useState(false);
  const router = useRouter();
  const { login } = useAuthStore();
```

```
  const {
    register,
    handleSubmit,
    formState: { errors },
    setError,
  } = useForm({
    resolver: zodResolver(loginSchema),
    defaultValues: {
      email: '',
      password: '',
      remember: false,
    },
  });
```

```
  const onSubmit = async (data: any) => {
    setIsLoading(true);
```

```

try {
  await login(data.email, data.password);
  logger.info('User logged in', { email: data.email });
  router.push('/home/overview');
} catch (error: any) {
  logger.error('Login failed', { error: error.message });
  setError('root', {
    type: 'manual',
    message: error.message || 'Invalid credentials',
  });
} finally {
  setIsLoading(false);
}
};

return (
  <div className="min-h-screen flex items-center justify-center bg-gradient-to-br
from-blue-50 to-green-50 dark:from-gray-900 dark:to-gray-800">
    <Card className="w-full max-w-md shadow-2xl border-0">
      <CardHeader className="space-y-1">
        <div className="flex items-center justify-center mb-4">
          <div className="bg-blue-600 p-3 rounded-full">
            <Lock className="h-8 w-8 text-white" />
          </div>
        </div>
        <CardTitle className="text-2xl font-bold text-center">
          Welcome Back
        </CardTitle>
        <CardDescription className="text-center">
          Enter your credentials to access your account
        </CardDescription>
      </CardHeader>
      <CardContent>
        <form onSubmit={handleSubmit(onSubmit)} className="space-y-4">
          <div className="space-y-2">
            <Label htmlFor="email">Email</Label>
            <div className="relative">
              <Mail className="absolute left-3 top-3 h-4 w-4 text-gray-400" />
              <Input
                id="email"
                type="email"
                placeholder="admin@agrodeep.com"
                className="pl-10"
                {...register('email')}
                disabled={isLoading}
              />
            </div>
          </div>
          {errors.email && (

```

```

    <p className="text-sm text-red-500">{errors.email.message}</p>
  )}
</div>

<div className="space-y-2">
  <div className="flex items-center justify-between">
    <Label htmlFor="password">Password</Label>
    <Link
      href="/forgot-password"
      className="text-sm text-blue-600 hover:underline"
    >
      Forgot password?
    </Link>
  </div>
  <div className="relative">
    <Lock className="absolute left-3 top-3 h-4 w-4 text-gray-400" />
    <Input
      id="password"
      type={showPassword ? 'text' : 'password'}
      placeholder="....."
      className="pl-10 pr-10"
      {...register('password')}
      disabled={isLoading}
    />
    <Button
      type="button"
      variant="ghost"
      size="sm"
      className="absolute right-0 top-0 h-full px-3"
      onClick={() => setShowPassword(!showPassword)}
    >
      {showPassword ? (
        <EyeOff className="h-4 w-4" />
      ) : (
        <Eye className="h-4 w-4" />
      )}
    </Button>
  </div>
  {errors.password && (
    <p className="text-sm text-red-500">{errors.password.message}</p>
  )}
</div>

<div className="flex items-center space-x-2">
  <Checkbox id="remember" {...register('remember')} />
  <Label htmlFor="remember" className="text-sm">
    Remember me
  </Label>
</div>

```

```

</div>

{errors.root && (
  <div className="bg-red-50 border border-red-200 text-red-700 px-4 py-3
rounded-md text-sm">
    {errors.root.message}
  </div>
)}

<Button
  type="submit"
  className="w-full bg-blue-600 hover:bg-blue-700"
  disabled={isLoading}
>
  {isLoading ? 'Signing in...' : 'Sign In'}
</Button>
</form>

<div className="relative my-6">
  <div className="absolute inset-0 flex items-center">
    <Separator />
  </div>
  <div className="relative flex justify-center text-xs uppercase">
    <span className="bg-background px-2 text-muted-foreground">
      Or continue with
    </span>
  </div>
</div>

<div className="grid grid-cols-2 gap-3">
  <Button variant="outline" type="button" disabled={isLoading}>
    <svg className="mr-2 h-4 w-4" viewBox="0 0 24 24">
      <path
        d="M22.56 12.25c0-.78-.07-1.53-.2-2.25H12v4.26h5.92c-.26 1.37-1.04 2.53-2.21
3.31v2.77h3.57c2.08-1.92 3.28-4.74 3.28-8.09z"
        fill="#4285F4"
      />
      <path
        d="M12 23c2.97 0 5.46-.98 7.28-2.66l-3.57-2.77c-.98.66-2.23 1.06-3.71
1.06-2.86 0-5.29-1.93-6.16-4.53H2.18v2.84C3.99 20.53 7.7 23 12 23z"
        fill="#34A853"
      />
      <path
        d="M5.84 14.09c-.22-.66-.35-1.36-.35-2.09s.13-1.43.35-2.09V7.07H2.18C1.43
8.55 1 10.22 1 12s.43 3.45 1.18 4.93l2.85-2.22.81-.62z"
        fill="#FBBC05"
      />
    </path>
  </div>

```

```

        d="M12 5.38c1.62 0 3.06.56 4.21 1.64l3.15-3.15C17.45 2.09 14.97 1 12 1 7.7 1
3.99 3.47 2.18 7.07l3.66 2.84c.87-2.6 3.3-4.53 6.16-4.53z"
        fill="#EA4335"
      />
    </svg>
    Google
  </Button>
  <Button variant="outline" type="button" disabled={isLoading}>
    <svg className="mr-2 h-4 w-4" fill="currentColor" viewBox="0 0 24 24">
      <path d="M24 12.073c0-6.627-5.373-12-12-12s-12 5.373-12 12c0 5.99 4.388
10.954 10.125 11.854v-8.385H7.078v-3.47h3.047V9.43c0-3.007 1.792-4.669 4.533-4.669
1.312 0 2.686.235 2.686.235v2.953H15.83c-1.491 0-1.956.925-1.956
1.874v2.25h3.328l-.532 3.47h-2.796v8.385C19.612 23.027 24 18.062 24 12.073z" />
    </svg>
    Facebook
  </Button>
</div>
</CardContent>
<CardFooter className="flex flex-col space-y-4">
  <div className="text-center text-sm">
    Don't have an account?{' '}
    <Link href="/register" className="text-blue-600 hover:underline font-medium">
      Sign up
    </Link>
  </div>
  <div className="text-center text-xs text-muted-foreground">
    By continuing, you agree to our{' '}
    <Link href="/terms" className="underline">
      Terms of Service
    </Link>{' '}
    and{' '}
    <Link href="/privacy" className="underline">
      Privacy Policy
    </Link>
  </div>
</CardFooter>
</Card>
</div>
);
}
...

```

```

### **`src/app/(auth)/register/page.tsx`**

```

```

````tsx

```

```

'use client';

```

```

import { useState } from 'react';

```

```

import { useRouter } from 'next/navigation';

```

```

import { zodResolver } from '@hookform/resolvers/zod';
import { useForm } from 'react-hook-form';
import Link from 'next/link';
import { Button } from '@components/ui/button';
import { Input } from '@components/ui/input';
import { Card, CardContent, CardDescription, CardFooter, CardHeader, CardTitle } from '@components/ui/card';
import { Label } from '@components/ui/label';
import { Checkbox } from '@components/ui/checkbox';
import { Select, SelectContent, SelectItem, SelectTrigger, SelectValue } from '@components/ui/select';
import { userSchema } from '@types/schema/users.schema';
import { api } from '@services/api';
import { logger } from '@services/monitoring/logger';
import { Eye, EyeOff, UserPlus, Mail, Lock, User, MapPin, Phone } from 'lucide-react';

```

```

export default function RegisterPage() {
  const [showPassword, setShowPassword] = useState(false);
  const [isLoading, setIsLoading] = useState(false);
  const router = useRouter();

```

```

  const {
    register,
    handleSubmit,
    formState: { errors },
    setError,
    watch,
  } = useForm({
    resolver: zodResolver(userSchema),
    defaultValues: {
      email: "",
      password: "",
      confirmPassword: "",
      firstName: "",
      lastName: "",
      phone: "",
      accountType: 'customer',
      terms: false,
    },
  });

```

```

  const password = watch('password');

```

```

  const onSubmit = async (data: any) => {
    if (data.password !== data.confirmPassword) {
      setError('confirmPassword', {
        type: 'manual',
        message: 'Passwords do not match',
      });
    }
  };

```

```

    });
    return;
}

setIsLoading(true);
try {
    await api.auth.register({
        email: data.email,
        password: data.password,
        firstName: data.firstName,
        lastName: data.lastName,
        phone: data.phone,
        accountType: data.accountType,
    });

    logger.info('User registered', { email: data.email });
    router.push('/register/success?email=' + encodeURIComponent(data.email));
} catch (error: any) {
    logger.error('Registration failed', { error: error.message });
    setError('root', {
        type: 'manual',
        message: error.message || 'Registration failed. Please try again.',
    });
} finally {
    setIsLoading(false);
}
};

return (
    <div className="min-h-screen flex items-center justify-center bg-gradient-to-br
from-blue-50 to-green-50 dark:from-gray-900 dark:to-gray-800 py-12 px-4">
        <Card className="w-full max-w-2xl shadow-2xl border-0">
            <CardHeader className="space-y-1">
                <div className="flex items-center justify-center mb-4">
                    <div className="bg-blue-600 p-3 rounded-full">
                        <UserPlus className="h-8 w-8 text-white" />
                    </div>
                </div>
                <CardTitle className="text-2xl font-bold text-center">
                    Create Your Account
                </CardTitle>
                <CardDescription className="text-center">
                    Join AgroDeep and transform your agricultural business
                </CardDescription>
            </CardHeader>
            <CardContent>
                <form onSubmit={handleSubmit(onSubmit)} className="space-y-6">
                    <div className="grid grid-cols-1 md:grid-cols-2 gap-4">

```

```

<div className="space-y-2">
  <Label htmlFor="firstName">First Name</Label>
  <div className="relative">
    <User className="absolute left-3 top-3 h-4 w-4 text-gray-400" />
    <Input
      id="firstName"
      placeholder="John"
      className="pl-10"
      {...register('firstName')}
      disabled={isLoading}
    />
  </div>
  {errors.firstName && (
    <p className="text-sm text-red-500">{errors.firstName.message}</p>
  )}
</div>

```

```

<div className="space-y-2">
  <Label htmlFor="lastName">Last Name</Label>
  <div className="relative">
    <User className="absolute left-3 top-3 h-4 w-4 text-gray-400" />
    <Input
      id="lastName"
      placeholder="Doe"
      className="pl-10"
      {...register('lastName')}
      disabled={isLoading}
    />
  </div>
  {errors.lastName && (
    <p className="text-sm text-red-500">{errors.lastName.message}</p>
  )}
</div>
</div>

```

```

<div className="space-y-2">
  <Label htmlFor="email">Email Address</Label>
  <div className="relative">
    <Mail className="absolute left-3 top-3 h-4 w-4 text-gray-400" />
    <Input
      id="email"
      type="email"
      placeholder="john@example.com"
      className="pl-10"
      {...register('email')}
      disabled={isLoading}
    />
  </div>
</div>

```



```

{errors.email && (
  <p className="text-sm text-red-500">{errors.email.message}</p>
)}
</div>

```

```

<div className="space-y-2">
  <Label htmlFor="phone">Phone Number</Label>
  <div className="relative">
    <Phone className="absolute left-3 top-3 h-4 w-4 text-gray-400" />
    <Input
      id="phone"
      type="tel"
      placeholder="+33 1 23 45 67 89"
      className="pl-10"
      {...register('phone')}
      disabled={isLoading}
    />
  </div>
  {errors.phone && (
    <p className="text-sm text-red-500">{errors.phone.message}</p>
  )}
</div>

```

```

<div className="space-y-2">
  <Label htmlFor="accountType">Account Type</Label>
  <Select {...register('accountType')} defaultValue="customer">
    <SelectTrigger>
      <SelectValue placeholder="Select account type" />
    </SelectTrigger>
    <SelectContent>
      <SelectItem value="customer">Customer (Buy Products)</SelectItem>
      <SelectItem value="farmer">Farmer (Sell Products)</SelectItem>
      <SelectItem value="logistics">Logistics Provider</SelectItem>
      <SelectItem value="distributor">Distributor</SelectItem>
    </SelectContent>
  </Select>
</div>

```

```

<div className="grid grid-cols-1 md:grid-cols-2 gap-4">
  <div className="space-y-2">
    <Label htmlFor="password">Password</Label>
    <div className="relative">
      <Lock className="absolute left-3 top-3 h-4 w-4 text-gray-400" />
      <Input
        id="password"
        type={showPassword ? 'text' : 'password'}
        placeholder="....."
        className="pl-10 pr-10"
      />
    </div>
  </div>

```

```

      {...register('password')}
      disabled={isLoading}
    />
    <Button
      type="button"
      variant="ghost"
      size="sm"
      className="absolute right-0 top-0 h-full px-3"
      onClick={() => setShowPassword(!showPassword)}
    >
      {showPassword ? (
        <EyeOff className="h-4 w-4" />
      ) : (
        <Eye className="h-4 w-4" />
      )}
    </Button>
  </div>
  {errors.password && (
    <p className="text-sm text-red-500">{errors.password.message}</p>
  )}
</div>

<div className="space-y-2">
  <Label htmlFor="confirmPassword">Confirm Password</Label>
  <div className="relative">
    <Lock className="absolute left-3 top-3 h-4 w-4 text-gray-400" />
    <Input
      id="confirmPassword"
      type={showPassword ? 'text' : 'password'}
      placeholder="....."
      className="pl-10"
      {...register('confirmPassword')}
      disabled={isLoading}
    />
  </div>
  {errors.confirmPassword && (
    <p className="text-sm text-red-500">{errors.confirmPassword.message}</p>
  )}
</div>
</div>

<div className="space-y-4">
  <div className="flex items-start space-x-2">
    <Checkbox id="terms" {...register('terms')} />
    <Label htmlFor="terms" className="text-sm leading-relaxed">
      I agree to the{' '}
      <Link href="/terms" className="text-blue-600 hover:underline">
        Terms of Service
    </Label>
  </div>

```

```

        </Link>{' '}
        and{' '}
        <Link href="/privacy" className="text-blue-600 hover:underline">
          Privacy Policy
        </Link>
      </Label>
    </div>
    {errors.terms && (
      <p className="text-sm text-red-500">You must accept the terms</p>
    )}

    <div className="flex items-start space-x-2">
      <Checkbox id="newsletter" />
      <Label htmlFor="newsletter" className="text-sm leading-relaxed">
        I want to receive updates about new products and features
      </Label>
    </div>
  </div>

  {errors.root && (
    <div className="bg-red-50 border border-red-200 text-red-700 px-4 py-3
rounded-md text-sm">
      {errors.root.message}
    </div>
  )}

  <Button
    type="submit"
    className="w-full bg-blue-600 hover:bg-blue-700"
    disabled={isLoading}
  >
    {isLoading ? 'Creating Account...' : 'Create Account'}
  </Button>
</form>
</CardContent>
<CardFooter className="flex flex-col space-y-4">
  <div className="text-center text-sm">
    Already have an account?{' '}
    <Link href="/login" className="text-blue-600 hover:underline font-medium">
      Sign in
    </Link>
  </div>
  <div className="text-center text-xs text-muted-foreground">
    AgroDeep helps you connect with farmers, distributors, and logistics
    providers across the agricultural ecosystem.
  </div>
</CardFooter>
</Card>

```

```
    </div>
  );
}
...
```

```
## 📁 **PAGES ADMINISTRATEUR**
```

```
### **`src/app/(admin)/overview/page.tsx`**
```

```
``tsx
```

```
'use client';
```

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

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

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

```
import { Tabs, TabsContent, TabsList, TabsTrigger } from '@/components/ui/tabs';
```

```
import {
```

```
  BarChart3,
```

```
  TrendingUp,
```

```
  Users,
```

```
  Package,
```

```
  ShoppingCart,
```

```
  DollarSign,
```

```
  Activity,
```

```
  ArrowUpRight,
```

```
  ArrowDownRight,
```

```
  Calendar,
```

```
  Download
```

```
} from 'lucide-react';
```

```
import { useAdminOverview } from '@/hooks/useAdmin';
```

```
import { formatCurrency, formatNumber } from '@/utils/formatters';
```

```
import { AreaChart, Area, BarChart, Bar, XAxis, YAxis, CartesianGrid, Tooltip,
ResponsiveContainer } from 'recharts';
```

```
import { Badge } from '@/components/ui/badge';
```

```
import { RecentOrdersTable } from '@/components/admin/RecentOrdersTable';
```

```
import { TopProductsChart } from '@/components/admin/TopProductsChart';
```

```
import { PlatformMetrics } from '@/components/admin/PlatformMetrics';
```

```
import { DateRangePicker } from '@/components/ui/date-range-picker';
```

```
export default function AdminOverviewPage() {
```

```
  const { metrics, loading, refreshMetrics } = useAdminOverview();
```

```
  const [dateRange, setDateRange] = useState({
```

```
    from: new Date(new Date().setDate(new Date().getDate() - 30)),
```

```
    to: new Date(),
```

```
  });
```

```
  const stats = [
```

```
    {
```

```

    title: 'Total Revenue',
    value: formatCurrency(metrics?.totalRevenue || 0),
    icon: DollarSign,
    trend: 12.5,
    color: 'text-green-600',
    bgColor: 'bg-green-50',
  },
  {
    title: 'Total Orders',
    value: formatNumber(metrics?.totalOrders || 0),
    icon: ShoppingCart,
    trend: 8.2,
    color: 'text-blue-600',
    bgColor: 'bg-blue-50',
  },
  {
    title: 'Active Users',
    value: formatNumber(metrics?.activeUsers || 0),
    icon: Users,
    trend: 15.3,
    color: 'text-purple-600',
    bgColor: 'bg-purple-50',
  },
  {
    title: 'Total Products',
    value: formatNumber(metrics?.totalProducts || 0),
    icon: Package,
    trend: 5.7,
    color: 'text-orange-600',
    bgColor: 'bg-orange-50',
  },
];

```

```

const revenueData = [
  { month: 'Jan', revenue: 42000, orders: 150 },
  { month: 'Feb', revenue: 52000, orders: 180 },
  { month: 'Mar', revenue: 61000, orders: 210 },
  { month: 'Apr', revenue: 78000, orders: 250 },
  { month: 'May', revenue: 85000, orders: 280 },
  { month: 'Jun', revenue: 92000, orders: 310 },
];

```

```

return (
  <div className="space-y-6">
    <div className="flex flex-col sm:flex-row justify-between items-start sm:items-center gap-4">
      <div>
        <h1 className="text-3xl font-bold tracking-tight">Dashboard Overview</h1>

```

```
<p className="text-muted-foreground">
  Welcome back, Administrator. Here's what's happening with your
platform today.
```

```
</p>
</div>
<div className="flex items-center gap-2">
  <DatePicker
    date={dateRange}
    onChange={setDateRange}
  />
  <Button variant="outline" size="sm">
    <Download className="h-4 w-4 mr-2" />
    Export
  </Button>
</div>
</div>
```

```
{/* Stats Grid */}
<div className="grid gap-4 md:grid-cols-2 lg:grid-cols-4">
  {stats.map((stat, index) => (
    <Card key={index}>
      <CardHeader className="flex flex-row items-center justify-between space-y-0
pb-2">
        <CardTitle className="text-sm font-medium">
          {stat.title}
        </CardTitle>
        <div className={`p-2 rounded-full ${stat.bgColor}`}>
          <stat.icon className={`h-4 w-4 ${stat.color}`} />
        </div>
      </CardHeader>
      <CardContent>
        <div className="text-2xl font-bold">{stat.value}</div>
        <div className="flex items-center text-xs text-muted-foreground mt-1">
          {stat.trend > 0 ? (
            <ArrowUpRight className="h-3 w-3 text-green-600 mr-1" />
          ) : (
            <ArrowDownRight className="h-3 w-3 text-red-600 mr-1" />
          )}
          <span className={stat.trend > 0 ? 'text-green-600' : 'text-red-600'}>
            {Math.abs(stat.trend)}%
          </span>
          <span className="ml-1">from last month</span>
        </div>
      </CardContent>
    </Card>
  )})
</div>
```

```

<Tabs defaultValue="overview" className="space-y-4">
  <TabsList>
    <TabsTrigger value="overview">Overview</TabsTrigger>
    <TabsTrigger value="analytics">Analytics</TabsTrigger>
    <TabsTrigger value="reports">Reports</TabsTrigger>
  </TabsList>

  <TabsContent value="overview" className="space-y-4">
    <div className="grid gap-4 md:grid-cols-2 lg:grid-cols-7">
      <Card className="col-span-4">
        <CardHeader>
          <CardTitle>Revenue Overview</CardTitle>
          <CardDescription>
            Total revenue and orders for the last 6 months
          </CardDescription>
        </CardHeader>
        <CardContent className="pl-2">
          <ResponsiveContainer width="100%" height={350}>
            <AreaChart data={revenueData}>
              <CartesianGrid strokeDasharray="3 3" />
              <XAxis dataKey="month" />
              <YAxis />
              <Tooltip
                formatter={(value) => [
                  formatCurrency(value as number),
                  'Revenue',
                ]}
              />
              <Area
                type="monotone"
                dataKey="revenue"
                stroke="#2563eb"
                fill="#93c5fd"
                fillOpacity={0.3}
              />
            </AreaChart>
          </ResponsiveContainer>
        </CardContent>
      </Card>

      <Card className="col-span-3">
        <CardHeader>
          <CardTitle>Top Products</CardTitle>
          <CardDescription>Best selling products this month</CardDescription>
        </CardHeader>
        <CardContent>
          <TopProductsChart />
        </CardContent>
      </Card>
    </div>
  </TabsContent>
</Tabs>

```

```
</Card>
</div>
```

```
<div className="grid gap-4 md:grid-cols-2 lg:grid-cols-3">
  <Card className="col-span-2">
    <CardHeader>
      <CardTitle>Recent Orders</CardTitle>
      <CardDescription>Latest orders that need attention</CardDescription>
    </CardHeader>
    <CardContent>
      <RecentOrdersTable />
    </CardContent>
  </Card>
```

```

  <Card>
    <CardHeader>
      <CardTitle>Platform Health</CardTitle>
      <CardDescription>System metrics and performance</CardDescription>
    </CardHeader>
    <CardContent>
      <PlatformMetrics />
    </CardContent>
  </Card>
</div>
</TabsContent>
```

```
<TabsContent value="analytics">
  <Card>
    <CardHeader>
      <CardTitle>Advanced Analytics</CardTitle>
      <CardDescription>
        Detailed insights and metrics for better decision making
      </CardDescription>
    </CardHeader>
    <CardContent>
      <div className="h-[400px]">
        <ResponsiveContainer width="100%" height="100%">
          <BarChart data={revenueData}>
            <CartesianGrid strokeDasharray="3 3" />
            <XAxis dataKey="month" />
            <YAxis />
            <Tooltip />
            <Bar dataKey="revenue" fill="#2563eb" name="Revenue" />
            <Bar dataKey="orders" fill="#10b981" name="Orders" />
          </BarChart>
        </ResponsiveContainer>
      </div>
    </CardContent>
  </Card>
```



```

        </Card>
      </TabsContent>
    </Tabs>

    { /* Quick Actions */ }
    <Card>
      <CardHeader>
        <CardTitle>Quick Actions</CardTitle>
        <CardDescription>Common administrative tasks</CardDescription>
      </CardHeader>
      <CardContent>
        <div className="grid grid-cols-2 md:grid-cols-4 gap-4">
          <Button variant="outline" className="h-auto py-4">
            <Package className="h-4 w-4 mr-2" />
            Add Product
          </Button>
          <Button variant="outline" className="h-auto py-4">
            <Users className="h-4 w-4 mr-2" />
            Manage Users
          </Button>
          <Button variant="outline" className="h-auto py-4">
            <DollarSign className="h-4 w-4 mr-2" />
            View Reports
          </Button>
          <Button variant="outline" className="h-auto py-4">
            <Activity className="h-4 w-4 mr-2" />
            System Logs
          </Button>
        </div>
      </CardContent>
    </Card>
  </div>
);
}
...

### **`src/app/(admin)/analytics/page.tsx`**
```tsx
'use client';

import { useState } from 'react';
import { Card, CardContent, CardDescription, CardHeader, CardTitle } from
'@/components/ui/card';
import { Tabs, TabsContent, TabsList, TabsTrigger } from '@/components/ui/tabs';
import { Button } from '@/components/ui/button';
import { Select, SelectContent, SelectItem, SelectTrigger, SelectValue } from
'@/components/ui/select';
import { DateRangePicker } from '@/components/ui/date-range-picker';

```

```

import {
  LineChart,
  BarChart,
  PieChart,
  TrendingUp,
  Download,
  Filter,
  Users,
  ShoppingBag,
  Globe,
  Smartphone,
  Calendar
} from 'lucide-react';
import {
  LineChart as RechartsLine,
  BarChart as RechartsBar,
  PieChart as RechartsPie,
  Line,
  Bar,
  Pie,
  Cell,
  XAxis,
  YAxis,
  CartesianGrid,
  Tooltip,
  Legend,
  ResponsiveContainer
} from 'recharts';
import { useAdminAnalytics } from '@/hooks/useAnalytics';
import { formatCurrency, formatNumber } from '@/utils/formatters';
import { Badge } from '@components/ui/badge';

const COLORS = ['#0088FE', '#00C49F', '#FFBB28', '#FF8042', '#8884D8'];

export default function AnalyticsPage() {
  const [dateRange, setDateRange] = useState({
    from: new Date(new Date().setMonth(new Date().getMonth() - 3)),
    to: new Date(),
  });

  const [metricType, setMetricType] = useState('revenue');
  const [granularity, setGranularity] = useState('monthly');

  const { analytics, loading } = useAdminAnalytics(dateRange);

  const revenueData = [
    { month: 'Jan', revenue: 42000, target: 45000 },
    { month: 'Feb', revenue: 52000, target: 50000 },
    { month: 'Mar', revenue: 61000, target: 55000 },
  ]

```

```

    { month: 'Apr', revenue: 78000, target: 60000 },
    { month: 'May', revenue: 85000, target: 65000 },
    { month: 'Jun', revenue: 92000, target: 70000 },
    { month: 'Jul', revenue: 88000, target: 75000 },
    { month: 'Aug', revenue: 95000, target: 80000 },
  ];

```

```

const categoryData = [
  { name: 'Vegetables', value: 35, color: '#10b981' },
  { name: 'Fruits', value: 25, color: '#3b82f6' },
  { name: 'Grains', value: 20, color: '#f59e0b' },
  { name: 'Dairy', value: 15, color: '#8b5cf6' },
  { name: 'Meat', value: 5, color: '#ef4444' },
];

```

```

const deviceData = [
  { device: 'Mobile', users: 65, sessions: 12000 },
  { device: 'Desktop', users: 25, sessions: 8000 },
  { device: 'Tablet', users: 10, sessions: 3000 },
];

```

```

const topProducts = [
  { name: 'Organic Tomatoes', sales: 1250, revenue: 18750 },
  { name: 'Fresh Apples', sales: 980, revenue: 14700 },
  { name: 'Whole Wheat', sales: 750, revenue: 11250 },
  { name: 'Fresh Milk', sales: 620, revenue: 9300 },
  { name: 'Free-range Eggs', sales: 540, revenue: 8100 },
];

```

```

return (
  <div className="space-y-6">
    <div className="flex flex-col lg:flex-row justify-between items-start lg:items-center gap-4">
      <div>
        <h1 className="text-3xl font-bold tracking-tight">Advanced Analytics</h1>
        <p className="text-muted-foreground">
          Deep insights and metrics for data-driven decisions
        </p>
      </div>
      <div className="flex flex-wrap items-center gap-2">
        <DateRangePicker
          date={dateRange}
          onChange={setDateRange}
        />
        <Select value={granularity} onChange={setGranularity}>
          <SelectTrigger className="w-[120px]">
            <SelectValue placeholder="Granularity" />
          </SelectTrigger>

```

```

    <SelectContent>
      <SelectItem value="daily">Daily</SelectItem>
      <SelectItem value="weekly">Weekly</SelectItem>
      <SelectItem value="monthly">Monthly</SelectItem>
      <SelectItem value="quarterly">Quarterly</SelectItem>
    </SelectContent>
  </Select>
  <Select value={metricType} onValueChange={setMetricType}>
    <SelectTrigger className="w-[140px]">
      <SelectValue placeholder="Metric Type" />
    </SelectTrigger>
    <SelectContent>
      <SelectItem value="revenue">Revenue</SelectItem>
      <SelectItem value="orders">Orders</SelectItem>
      <SelectItem value="users">Users</SelectItem>
      <SelectItem value="products">Products</SelectItem>
    </SelectContent>
  </Select>
  <Button variant="outline" size="sm">
    <Filter className="h-4 w-4 mr-2" />
    Filters
  </Button>
  <Button variant="outline" size="sm">
    <Download className="h-4 w-4 mr-2" />
    Export
  </Button>
</div>
</div>

{/* KPIs */}
<div className="grid gap-4 md:grid-cols-2 lg:grid-cols-4">
  <Card>
    <CardHeader className="flex flex-row items-center justify-between space-y-0 pb-2">
      <CardTitle className="text-sm font-medium">Total Revenue</CardTitle>
      <TrendingUp className="h-4 w-4 text-muted-foreground" />
    </CardHeader>
    <CardContent>
      <div className="text-2xl font-bold">{formatCurrency(125000)}</div>
      <p className="text-xs text-muted-foreground">
        +15.2% from last period
      </p>
    </CardContent>
  </Card>
  <Card>
    <CardHeader className="flex flex-row items-center justify-between space-y-0 pb-2">
      <CardTitle className="text-sm font-medium">Conversion Rate</CardTitle>
      <Users className="h-4 w-4 text-muted-foreground" />
    </CardHeader>

```

```

<CardContent>
  <div className="text-2xl font-bold">3.2%</div>
  <p className="text-xs text-muted-foreground">
    +0.8% from last period
  </p>
</CardContent>
</Card>
<Card>
  <CardHeader className="flex flex-row items-center justify-between space-y-0 pb-2">
    <CardTitle className="text-sm font-medium">Avg Order Value</CardTitle>
    <ShoppingBag className="h-4 w-4 text-muted-foreground" />
  </CardHeader>
  <CardContent>
    <div className="text-2xl font-bold">{formatCurrency(85.50)}</div>
    <p className="text-xs text-muted-foreground">
      +5.3% from last period
    </p>
  </CardContent>
</Card>
<Card>
  <CardHeader className="flex flex-row items-center justify-between space-y-0 pb-2">
    <CardTitle className="text-sm font-medium">Customer Retention</CardTitle>
    <Globe className="h-4 w-4 text-muted-foreground" />
  </CardHeader>
  <CardContent>
    <div className="text-2xl font-bold">78.5%</div>
    <p className="text-xs text-muted-foreground">
      +2.1% from last period
    </p>
  </CardContent>
</Card>
</div>

<Tabs defaultValue="revenue" className="space-y-4">
  <TabsList>
    <TabsTrigger value="revenue">Revenue Analytics</TabsTrigger>
    <TabsTrigger value="products">Product Analytics</TabsTrigger>
    <TabsTrigger value="users">User Analytics</TabsTrigger>
    <TabsTrigger value="geography">Geography</TabsTrigger>
  </TabsList>

  <TabsContent value="revenue" className="space-y-4">
    <div className="grid gap-4 md:grid-cols-2 lg:grid-cols-7">
      <Card className="col-span-4">
        <CardHeader>
          <CardTitle>Revenue Trend</CardTitle>
          <CardDescription>
            Revenue performance vs targets over time

```

```

    </CardDescription>
  </CardHeader>
  <CardContent className="pl-2">
    <ResponsiveContainer width="100%" height={350}>
      <RechartsLine data={revenueData}>
        <CartesianGrid strokeDasharray="3 3" />
        <XAxis dataKey="month" />
        <YAxis />
        <Tooltip formatter={(value) => [formatCurrency(value as number), 'Value']} />
        <Legend />
        <Line
          type="monotone"
          dataKey="revenue"
          stroke="#2563eb"
          strokeWidth={2}
          dot={{ r: 4 }}
          activeDot={{ r: 6 }}
        />
        <Line
          type="monotone"
          dataKey="target"
          stroke="#10b981"
          strokeWidth={2}
          strokeDasharray="5 5"
        />
      </RechartsLine>
    </ResponsiveContainer>
  </CardContent>
</Card>

```

```

<Card className="col-span-3">
  <CardHeader>
    <CardTitle>Revenue by Category</CardTitle>
    <CardDescription>Distribution of revenue across product
categories</CardDescription>
  </CardHeader>
  <CardContent>
    <div className="h-[300px]">
      <ResponsiveContainer width="100%" height="100%">
        <RechartsPie>
          <Pie
            data={categoryData}
            cx="50%"
            cy="50%"
            labelLine={false}
            label={({ name, percent }) => `${name}: ${(percent * 100).toFixed(0)}%`}
            outerRadius={80}
            fill="#8884d8"

```

```

        dataKey="value"
      >
        {categoryData.map((entry, index) => (
          <Cell key={`cell-${index}`} fill={entry.color} />
        ))}
      </Pie>
      <Tooltip formatter={(value) => [`${value}%`, 'Percentage']} />
    </RechartsPie>
  </ResponsiveContainer>
</div>
</CardContent>
</Card>
</div>
</TabsContent>

<TabsContent value="products" className="space-y-4">
  <div className="grid gap-4 md:grid-cols-2">
    <Card>
      <CardHeader>
        <CardTitle>Top Performing Products</CardTitle>
        <CardDescription>Best sellers by revenue and units sold</CardDescription>
      </CardHeader>
      <CardContent>
        <div className="space-y-4">
          {topProducts.map((product, index) => (
            <div key={index} className="flex items-center justify-between">
              <div className="space-y-1">
                <p className="font-medium">{product.name}</p>
                <div className="flex items-center gap-2 text-sm text-muted-foreground">
                  <span>{formatNumber(product.sales)} units</span>
                  <span>•</span>
                  <span>{formatCurrency(product.revenue)}</span>
                </div>
              </div>
              <Badge variant="outline">{index + 1}</Badge>
            </div>
          ))}
        </div>
      </CardContent>
    </Card>

    <Card>
      <CardHeader>
        <CardTitle>Product Performance</CardTitle>
        <CardDescription>Sales distribution across products</CardDescription>
      </CardHeader>
      <CardContent>
        <div className="h-[300px]">

```

```

    <ResponsiveContainer width="100%" height="100%">
      <RechartsBar data={topProducts}>
        <CartesianGrid strokeDasharray="3 3" />
        <XAxis dataKey="name" angle={-45} textAnchor="end" height={60} />
        <YAxis />
        <Tooltip formatter={(value) => [formatCurrency(value as number), 'Revenue']}>
          <Bar dataKey="revenue" fill="#2563eb" />
        </RechartsBar>
      </ResponsiveContainer>
    </div>
  </CardContent>
</Card>
</div>
</TabsContent>

<TabsContent value="users" className="space-y-4">
  <div className="grid gap-4 md:grid-cols-2">
    <Card>
      <CardHeader>
        <CardTitle>User Devices</CardTitle>
        <CardDescription>Platform usage by device type</CardDescription>
      </CardHeader>
      <CardContent>
        <div className="space-y-4">
          {deviceData.map((device, index) => (
            <div key={index} className="space-y-2">
              <div className="flex items-center justify-between">
                <div className="flex items-center gap-2">
                  <Smartphone className="h-4 w-4" />
                  <span className="font-medium">{device.device}</span>
                </div>
                <span className="text-sm text-muted-foreground">
                  {device.users}% of users
                </span>
              </div>
              <div className="w-full bg-gray-100 rounded-full h-2">
                <div
                  className="bg-blue-600 h-2 rounded-full"
                  style={{ width: `${device.users}%` }}
                />
              </div>
            </div>
          ))}
        </div>
      </CardContent>
    </Card>
  </div>

```



```

<Card>
  <CardHeader>
    <CardTitle>User Growth</CardTitle>
    <CardDescription>New user acquisition over time</CardDescription>
  </CardHeader>
  <CardContent>
    <div className="h-[300px]">
      <ResponsiveContainer width="100%" height="100%">
        <RechartsBar
          data={[
            { month: 'Jan', newUsers: 150, returningUsers: 450 },
            { month: 'Feb', newUsers: 180, returningUsers: 520 },
            { month: 'Mar', newUsers: 210, returningUsers: 580 },
            { month: 'Apr', newUsers: 250, returningUsers: 620 },
            { month: 'May', newUsers: 280, returningUsers: 680 },
            { month: 'Jun', newUsers: 310, returningUsers: 720 },
          ]}
        >
          <CartesianGrid strokeDasharray="3 3" />
          <XAxis dataKey="month" />
          <YAxis />
          <Tooltip />
          <Legend />
          <Bar dataKey="newUsers" fill="#10b981" name="New Users" />
          <Bar dataKey="returningUsers" fill="#2563eb" name="Returning Users" />
        </RechartsBar>
      </ResponsiveContainer>
    </div>
  </CardContent>
</Card>
</div>
</TabsContent>
</Tabs>

```

```

{/* Advanced Analytics Features */}
<Card>
  <CardHeader>
    <CardTitle>Advanced Metrics</CardTitle>
    <CardDescription>Detailed platform performance indicators</CardDescription>
  </CardHeader>
  <CardContent>
    <div className="grid gap-6 md:grid-cols-2 lg:grid-cols-3">
      <div className="space-y-2">
        <h4 className="font-medium">Customer Lifetime Value (CLV)</h4>
        <div className="text-2xl font-bold">{formatCurrency(1250)}</div>
        <p className="text-sm text-muted-foreground">Average per customer</p>
      </div>
      <div className="space-y-2">

```

```

        <h4 className="font-medium">Churn Rate</h4>
        <div className="text-2xl font-bold">2.1%</div>
        <p className="text-sm text-muted-foreground">Monthly customer churn</p>
      </div>
      <div className="space-y-2">
        <h4 className="font-medium">Customer Acquisition Cost</h4>
        <div className="text-2xl font-bold">{formatCurrency(45.80)}</div>
        <p className="text-sm text-muted-foreground">Average acquisition cost</p>
      </div>
    </div>
  </CardContent>
</Card>
</div>
);
}
...

```

```

### **`src/app/(admin)/inventory/page.tsx`**

```

```

```.tsx

```

```

'use client';

```

```

import { 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 { Table, TableBody, TableCell, TableHead, TableHeader, TableRow } from
'@/components/ui/table';
import { Badge } from '@components/ui/badge';
import { Tabs, TabsContent, TabsList, TabsTrigger } from '@components/ui/tabs';
import {
  Package,
  Plus,
  Search,
  Filter,
  AlertTriangle,
  CheckCircle,
  TrendingDown,
  Download,
  RefreshCw,
  BarChart3,
  ShoppingCart
} from 'lucide-react';
import { useInventory } from '@/hooks/useInventory';
import { formatCurrency, formatNumber } from '@/utils/formatters';
import { InventoryDialog } from '@components/admin/InventoryDialog';
import { Alert, AlertDescription, AlertTitle } from '@components/ui/alert';
import { Progress } from '@components/ui/progress';

```

```

import { Select, SelectContent, SelectItem, SelectTrigger, SelectValue } from
'@/components/ui/select';

export default function InventoryPage() {
  const [searchQuery, setSearchQuery] = useState("");
  const [categoryFilter, setCategoryFilter] = useState('all');
  const [stockFilter, setStockFilter] = useState('all');
  const [showAddDialog, setShowAddDialog] = useState(false);

  const { inventory, loading, lowStockItems, refetch } = useInventory();

  // Filter inventory based on search and filters
  const filteredInventory = inventory.filter(item => {
    const matchesSearch =
item.product.name.toLowerCase().includes(searchQuery.toLowerCase()) ||
    item.product.sku?.toLowerCase().includes(searchQuery.toLowerCase());
    const matchesCategory = categoryFilter === 'all' || item.product.category ===
categoryFilter;
    const matchesStock = stockFilter === 'all' ||
        (stockFilter === 'low' && item.quantity <= item.reorderPoint) ||
        (stockFilter === 'out' && item.quantity === 0) ||
        (stockFilter === 'healthy' && item.quantity > item.reorderPoint);

    return matchesSearch && matchesCategory && matchesStock;
  });

  const inventoryValue = inventory.reduce((total, item) => {
    return total + (item.product.price * item.quantity);
  }, 0);

  const categories = ['Vegetables', 'Fruits', 'Grains', 'Dairy', 'Meat', 'Beverages'];

  return (
    <div className="space-y-6">
      <div className="flex flex-col lg:flex-row justify-between items-start lg:items-center
gap-4">
        <div>
          <h1 className="text-3xl font-bold tracking-tight">Inventory Management</h1>
          <p className="text-muted-foreground">
            Monitor and manage your product inventory in real-time
          </p>
        </div>
        <div className="flex items-center gap-2">
          <Button variant="outline" size="sm" onClick={refetch}>
            <RefreshCw className="h-4 w-4 mr-2" />
            Refresh
          </Button>
          <Button variant="outline" size="sm">

```

```

        <Download className="h-4 w-4 mr-2" />
        Export
    </Button>
    <Button onClick={() => setShowAddDialog(true)}>
        <Plus className="h-4 w-4 mr-2" />
        Add Product
    </Button>
</div>
</div>

{/* Stats Overview */}
<div className="grid gap-4 md:grid-cols-2 lg:grid-cols-4">
    <Card>
        <CardHeader className="flex flex-row items-center justify-between space-y-0 pb-2">
            <CardTitle className="text-sm font-medium">Total Products</CardTitle>
            <Package className="h-4 w-4 text-muted-foreground" />
        </CardHeader>
        <CardContent>
            <div className="text-2xl font-bold">{formatNumber(inventory.length)}</div>
            <p className="text-xs text-muted-foreground">
                Active products in inventory
            </p>
        </CardContent>
    </Card>
    <Card>
        <CardHeader className="flex flex-row items-center justify-between space-y-0 pb-2">
            <CardTitle className="text-sm font-medium">Total Value</CardTitle>
            <TrendingDown className="h-4 w-4 text-muted-foreground" />
        </CardHeader>
        <CardContent>
            <div className="text-2xl font-bold">{formatCurrency(inventoryValue)}</div>
            <p className="text-xs text-muted-foreground">
                Current inventory worth
            </p>
        </CardContent>
    </Card>
    <Card>
        <CardHeader className="flex flex-row items-center justify-between space-y-0 pb-2">
            <CardTitle className="text-sm font-medium">Low Stock</CardTitle>
            <AlertTriangle className="h-4 w-4 text-muted-foreground" />
        </CardHeader>
        <CardContent>
            <div className="text-2xl font-bold">{lowStockItems.length}</div>
            <p className="text-xs text-muted-foreground">
                Products need restocking
            </p>
        </CardContent>
    </Card>

```

```

<Card>
  <CardHeader className="flex flex-row items-center justify-between space-y-0 pb-2">
    <CardTitle className="text-sm font-medium">Out of Stock</CardTitle>
    <ShoppingCart className="h-4 w-4 text-muted-foreground" />
  </CardHeader>
  <CardContent>
    <div className="text-2xl font-bold">
      {inventory.filter(item => item.quantity === 0).length}
    </div>
    <p className="text-xs text-muted-foreground">
      Products unavailable
    </p>
  </CardContent>
</Card>
</div>

{/* Alerts */}
{lowStockItems.length > 0 && (
  <Alert variant="destructive">
    <AlertTriangle className="h-4 w-4" />
    <AlertTitle>Low Stock Alert</AlertTitle>
    <AlertDescription>
      {lowStockItems.length} product{lowStockItems.length > 1 ? 's' : ''}
      {lowStockItems.length > 1 ? 'are' : 'is'} running low on stock. Consider restocking soon.
    </AlertDescription>
  </Alert>
)}

<Tabs defaultValue="all" className="space-y-4">
  <div className="flex flex-col lg:flex-row justify-between items-start lg:items-center gap-4">
    <TabsList>
      <TabsTrigger value="all">All Products</TabsTrigger>
      <TabsTrigger value="low">Low Stock</TabsTrigger>
      <TabsTrigger value="out">Out of Stock</TabsTrigger>
      <TabsTrigger value="analytics">Analytics</TabsTrigger>
    </TabsList>

    <div className="flex flex-col sm:flex-row gap-2 w-full lg:w-auto">
      <div className="relative flex-1 sm:flex-none">
        <Search className="absolute left-3 top-3 h-4 w-4 text-gray-400" />
        <Input
          placeholder="Search products..."
          value={searchQuery}
          onChange={(e) => setSearchQuery(e.target.value)}
          className="pl-10 w-full sm:w-[250px]"
        />
      </div>
    </div>
  </div>

```

```

<Select value={categoryFilter} onValueChange={setCategoryFilter}>
  <SelectTrigger className="w-full sm:w-[180px]">
    <SelectValue placeholder="Category" />
  </SelectTrigger>
  <SelectContent>
    <SelectItem value="all">All Categories</SelectItem>
    {categories.map(category => (
      <SelectItem key={category} value={category}>{category}</SelectItem>
    ))}
  </SelectContent>
</Select>
<Select value={stockFilter} onValueChange={setStockFilter}>
  <SelectTrigger className="w-full sm:w-[180px]">
    <SelectValue placeholder="Stock Status" />
  </SelectTrigger>
  <SelectContent>
    <SelectItem value="all">All Status</SelectItem>
    <SelectItem value="low">Low Stock</SelectItem>
    <SelectItem value="out">Out of Stock</SelectItem>
    <SelectItem value="healthy">Healthy</SelectItem>
  </SelectContent>
</Select>
</div>
</div>

```

```

<TabsContent value="all" className="space-y-4">
  <Card>
    <CardHeader>
      <CardTitle>Product Inventory</CardTitle>
      <CardDescription>
        Manage your product stock levels and details
      </CardDescription>
    </CardHeader>
    <CardContent>
      <Table>
        <TableHeader>
          <TableRow>
            <TableHead>Product</TableHead>
            <TableHead>SKU</TableHead>
            <TableHead>Category</TableHead>
            <TableHead>Current Stock</TableHead>
            <TableHead>Reorder Point</TableHead>
            <TableHead>Status</TableHead>
            <TableHead>Value</TableHead>
            <TableHead className="text-right">Actions</TableHead>
          </TableRow>
        </TableHeader>
        <TableBody>

```

```

{filteredInventory.map((item) => (
  <TableRow key={item.id}>
    <TableCell className="font-medium">
      <div className="flex items-center gap-3">
        {item.product.image} && (
          <img
            src={item.product.image}
            alt={item.product.name}
            className="h-10 w-10 rounded-md object-cover"
          />
        )}
      <div>
        <p>{item.product.name}</p>
        <p className="text-sm text-muted-foreground">
          {formatCurrency(item.product.price)} / unit
        </p>
      </div>
    </div>
  </TableCell>
  <TableCell>{item.product.sku || 'N/A'}</TableCell>
  <TableCell>
    <Badge variant="outline">{item.product.category}</Badge>
  </TableCell>
  <TableCell>
    <div className="space-y-1">
      <div className="flex justify-between text-sm">
        <span>{item.quantity} units</span>
        <span className="text-muted-foreground">
          {item.reservedQuantity > 0 && `${item.reservedQuantity} reserved`}
        </span>
      </div>
      <Progress
        value={(item.quantity / (item.reorderPoint * 3)) * 100}
        className="h-2"
      />
    </div>
  </TableCell>
  <TableCell>{item.reorderPoint}</TableCell>
  <TableCell>
    {item.quantity === 0 ? (
      <Badge variant="destructive">Out of Stock</Badge>
    ) : item.quantity <= item.reorderPoint ? (
      <Badge variant="warning">Low Stock</Badge>
    ) : (
      <Badge variant="success">In Stock</Badge>
    )}
  </TableCell>
  <TableCell className="font-medium">

```

```

        {formatCurrency(item.product.price * item.quantity)}
      </TableCell>
      <TableCell className="text-right">
        <div className="flex justify-end gap-2">
          <Button variant="outline" size="sm">
            Adjust
          </Button>
          <Button variant="outline" size="sm">
            View
          </Button>
        </div>
      </TableCell>
    </TableRow>
  )})
</TableBody>
</Table>
</CardContent>
</Card>
</TabsContent>

<TabsContent value="low">
  <Card>
    <CardHeader>
      <CardTitle>Low Stock Products</CardTitle>
      <CardDescription>
        Products that need immediate attention
      </CardDescription>
    </CardHeader>
    <CardContent>
      {lowStockItems.length > 0 ? (
        <Table>
          <TableHeader>
            <TableRow>
              <TableHead>Product</TableHead>
              <TableHead>Current Stock</TableHead>
              <TableHead>Reorder Point</TableHead>
              <TableHead>Days of Supply</TableHead>
              <TableHead className="text-right">Actions</TableHead>
            </TableRow>
          </TableHeader>
          <TableBody>
            {lowStockItems.map((item) => (
              <TableRow key={item.id}>
                <TableCell className="font-medium">
                  <div className="flex items-center gap-3">
                    {item.product.image && (
                      <img
                        src={item.product.image}

```



```

        alt={item.product.name}
        className="h-10 w-10 rounded-md object-cover"
      />
    ))
    <span>{item.product.name}</span>
  </div>
</TableCell>
<TableCell>
  <span className="font-medium text-red-600">{item.quantity}</span> units
</TableCell>
<TableCell>{item.reorderPoint} units</TableCell>
<TableCell>
  <Badge variant="outline">~3 days</Badge>
</TableCell>
<TableCell className="text-right">
  <Button size="sm">Order More</Button>
</TableCell>
</TableRow>
)))
</TableBody>
</Table>
) : (
  <div className="text-center py-8">
    <CheckCircle className="h-12 w-12 text-green-500 mx-auto mb-4" />
    <h3 className="text-lg font-medium">All products are well stocked!</h3>
    <p className="text-muted-foreground">
      No products are currently running low on inventory.
    </p>
  </div>
  )}
</CardContent>
</Card>
</TabsContent>

<TabsContent value="analytics">
  <div className="grid gap-4 md:grid-cols-2">
    <Card>
      <CardHeader>
        <CardTitle>Inventory Turnover</CardTitle>
        <CardDescription>
          How quickly inventory is sold and replaced
        </CardDescription>
      </CardHeader>
      <CardContent>
        <div className="space-y-4">
          <div className="flex items-center justify-between">
            <span className="text-sm">Vegetables</span>
            <span className="font-medium">8.2x</span>
          </div>

```

```

    </div>
    <div className="flex items-center justify-between">
      <span className="text-sm">Fruits</span>
      <span className="font-medium">6.5x</span>
    </div>
    <div className="flex items-center justify-between">
      <span className="text-sm">Dairy</span>
      <span className="font-medium">4.3x</span>
    </div>
    <div className="flex items-center justify-between">
      <span className="text-sm">Grains</span>
      <span className="font-medium">3.8x</span>
    </div>
  </div>
</CardContent>
</Card>
<Card>
  <CardHeader>
    <CardTitle>Stock Movement</CardTitle>
    <CardDescription>
      Recent inventory adjustments
    </CardDescription>
  </CardHeader>
  <CardContent>
    <div className="space-y-3">
      {[
        { product: 'Organic Tomatoes', change: '+500', reason: 'Restock', date: '2 hours
ago' },
        { product: 'Fresh Apples', change: '-250', reason: 'Sale', date: '4 hours ago' },
        { product: 'Whole Wheat', change: '+1000', reason: 'New shipment', date: '1 day
ago' },
        { product: 'Fresh Milk', change: '-120', reason: 'Sale', date: '2 days ago' },
      ].map((movement, index) => (
        <div key={index} className="flex items-center justify-between">
          <div>
            <p className="font-medium">{movement.product}</p>
            <p className="text-sm text-muted-foreground">{movement.reason}</p>
          </div>
          <div className="text-right">
            <p className={`font-medium ${movement.change.startsWith('+') ?
'text-green-600' : 'text-red-600'}`}>
              {movement.change} units
            </p>
            <p className="text-sm text-muted-foreground">{movement.date}</p>
          </div>
        </div>
      )))
    </div>
  </CardContent>
</Card>

```

```

        </CardContent>
      </Card>
    </div>
  </TabsContent>
</Tabs>

{/* Quick Actions */}
<div className="grid gap-4 md:grid-cols-2 lg:grid-cols-4">
  <Card>
    <CardHeader className="pb-2">
      <CardTitle className="text-sm font-medium">Quick Actions</CardTitle>
    </CardHeader>
    <CardContent>
      <div className="space-y-2">
        <Button variant="outline" className="w-full justify-start">
          <Plus className="h-4 w-4 mr-2" />
          Bulk Import
        </Button>
        <Button variant="outline" className="w-full justify-start">
          <Download className="h-4 w-4 mr-2" />
          Export Report
        </Button>
        <Button variant="outline" className="w-full justify-start">
          <BarChart3 className="h-4 w-4 mr-2" />
          Stock Forecast
        </Button>
        <Button variant="outline" className="w-full justify-start">
          <AlertTriangle className="h-4 w-4 mr-2" />
          Set Alerts
        </Button>
      </div>
    </CardContent>
  </Card>
</div>

```

```

    <InventoryDialog open={showAddDialog} onOpenChange={setShowAddDialog} />
  </div>

```

```

  );
}
...

```

```
## 📁 **PAGES MARKETPLACE**
```

```
### **`src/app/(market)/products/[uid]/page.tsx`**
```

```
``tsx
```

```
'use client';
```

```
import { useState } from 'react';
```

```

import { notFound } from 'next/navigation';
import { Button } from '@components/ui/button';
import { Card, CardContent } from '@components/ui/card';
import { Tabs, TabsContent, TabsList, TabsTrigger } from '@components/ui/tabs';
import { Badge } from '@components/ui/badge';
import { Separator } from '@components/ui/separator';
import {
  Star,
  ShoppingCart,
  Heart,
  Share2,
  Truck,
  Shield,
  Leaf,
  Calendar,
  MapPin,
  Users,
  Package
} from 'lucide-react';
import { useProduct } from '@hooks/useProducts';
import { formatCurrency } from '@utils/formatters';
import { useCartStore } from '@stores/useCartStore';
import { toast } from '@components/ui/use-toast';
import ImageGallery from '@components/market/products/ProductImages';
import ProductSuggestions from '@components/market/products/ProductSuggestions';
import CommentSection from '@components/market/products/CommentSection';
import RatingStars from '@components/market/products/RatingStars';

interface ProductPageProps {
  params: {
    uid: string;
  };
}

export default function ProductPage({ params }: ProductPageProps) {
  const { uid } = params;
  const { product, loading, error } = useProduct(uid);
  const [quantity, setQuantity] = useState(1);
  const [selectedImage, setSelectedImage] = useState(0);
  const [isWishlisted, setIsWishlisted] = useState(false);
  const addToCart = useCartStore((state) => state.addItem);

  if (loading) {
    return (
      <div className="flex items-center justify-center min-h-[400px]">
        <div className="text-center">
          <div className="animate-spin rounded-full h-12 w-12 border-b-2 border-blue-600
mx-auto"></div>

```

```

        <p className="mt-4 text-muted-foreground">Loading product details...</p>
      </div>
    </div>
  );
}

if (error || !product) {
  notFound();
}

const handleAddToCart = () => {
  addToCart({
    id: product.id,
    name: product.name,
    price: product.price,
    image: product.images?.[0],
    quantity,
  });

  toast({
    title: "Added to cart",
    description: `${quantity} × ${product.name} added to your cart`,
  });
};

const handleBuyNow = () => {
  handleAddToCart();
  // Navigate to checkout
};

const features = [
  { icon: Leaf, label: '100% Organic', description: 'Certified organic farming' },
  { icon: Shield, label: 'Quality Guarantee', description: '30-day freshness guarantee' },
  { icon: Truck, label: 'Fast Delivery', description: 'Next-day delivery available' },
  { icon: Calendar, label: 'Seasonal', description: 'Freshly harvested this season' },
];

return (
  <div className="min-h-screen bg-gray-50 dark:bg-gray-900">
    <div className="container mx-auto px-4 py-8">
      {/* Breadcrumb */}
      <div className="mb-6 text-sm text-muted-foreground">
        <span>Home</span>
        <span className="mx-2"></span>
        <span>Products</span>
        <span className="mx-2"></span>
        <span>{product.category}</span>
        <span className="mx-2"></span>

```

```
<span className="text-foreground font-medium">{product.name}</span>
</div>
```

```
<div className="grid gap-8 lg:grid-cols-2">
```

```
  {/ * Left Column - Images */}
```

```
  <div className="space-y-4">
```

```
    <ImageGallery
```

```
      images={product.images || []}
```

```
      selected={selectedImage}
```

```
      onSelect={setSelectedImage}
```

```
    />
```

```
  {/ * Product Tags */}
```

```
  <div className="flex flex-wrap gap-2">
```

```
    <Badge variant="secondary" className="bg-green-100 text-green-800">
```

```
      <Leaf className="h-3 w-3 mr-1" />
```

```
      Organic
```

```
    </Badge>
```

```
    <Badge variant="secondary">Seasonal</Badge>
```

```
    <Badge variant="secondary">Local</Badge>
```

```
    <Badge variant="secondary">Fresh</Badge>
```

```
    {product.tags?.map((tag) => (
```

```
      <Badge key={tag} variant="outline">
```

```
        {tag}
```

```
      </Badge>
```

```
    )))
```

```
  </div>
```

```
</div>
```

```
{/ * Right Column - Product Info */}
```

```
<div className="space-y-6">
```

```
  <div>
```

```
    <h1 className="text-3xl font-bold tracking-tight">{product.name}</h1>
```

```
    <div className="flex items-center gap-4 mt-2">
```

```
      <div className="flex items-center">
```

```
        <RatingStars rating={product.rating || 4.5} />
```

```
        <span className="ml-2 text-sm text-muted-foreground">
```

```
          ({product.reviewCount || 124} reviews)
```

```
        </span>
```

```
      </div>
```

```
      <Badge variant="outline">SKU: {product.sku || 'AGR-' + product.id}</Badge>
```

```
    </div>
```

```
  </div>
```

```
{/ * Price */}
```

```
<div className="space-y-2">
```

```
  <div className="flex items-center gap-3">
```

```
    <span className="text-3xl font-bold text-blue-600">
```

```

        {formatCurrency(product.price)}
      </span>
      {product.originalPrice && (
        <>
          <span className="text-lg text-muted-foreground line-through">
            {formatCurrency(product.originalPrice)}
          </span>
          <Badge variant="destructive" className="ml-2">
            Save {Math.round((1 - product.price / product.originalPrice) * 100)}%
          </Badge>
        </>
      )}
    </div>
    <p className="text-sm text-muted-foreground">
      Price per {product.unit || 'kg'} • Tax included
    </p>
  </div>

  {/* Description */}
  <div>
    <h3 className="text-lg font-semibold mb-2">Description</h3>
    <p className="text-gray-600 dark:text-gray-300">{product.description}</p>
  </div>

  {/* Key Features */}
  <div className="grid grid-cols-2 gap-4">
    {features.map((feature, index) => (
      <div key={index} className="flex items-start gap-3">
        <div className="bg-blue-50 p-2 rounded-full">
          <feature.icon className="h-5 w-5 text-blue-600" />
        </div>
        <div>
          <p className="font-medium">{feature.label}</p>
          <p className="text-sm text-muted-foreground">{feature.description}</p>
        </div>
      </div>
    ))}
  </div>

  {/* Quantity & Actions */}
  <div className="space-y-4">
    <div>
      <label className="block text-sm font-medium mb-2">Quantity</label>
      <div className="flex items-center gap-4">
        <div className="flex items-center border rounded-lg">
          <Button
            variant="ghost"
            size="icon"

```

```

        onClick={() => setQuantity(Math.max(1, quantity - 1))}
        disabled={quantity <= 1}
      >
        -
      </Button>
      <span className="w-12 text-center font-medium">{quantity}</span>
      <Button
        variant="ghost"
        size="icon"
        onClick={() => setQuantity(quantity + 1)}
      >
        +
      </Button>
    </div>
    <div className="text-sm text-muted-foreground">
      {product.stock || 50} units available
    </div>
  </div>
</div>

<div className="flex flex-col sm:flex-row gap-3">
  <Button
    size="lg"
    className="flex-1 bg-blue-600 hover:bg-blue-700"
    onClick={handleAddToCart}
  >
    <ShoppingCart className="mr-2 h-5 w-5" />
    Add to Cart
  </Button>
  <Button
    size="lg"
    variant="outline"
    className="flex-1"
    onClick={handleBuyNow}
  >
    Buy Now
  </Button>
  <Button
    size="icon"
    variant="outline"
    onClick={() => setIsWishlisted(!isWishlisted)}
  >
    <Heart className={`h-5 w-5 ${isWishlisted ? 'fill-red-500 text-red-500' : ''}`} />
  </Button>
  <Button size="icon" variant="outline">
    <Share2 className="h-5 w-5" />
  </Button>
</div>

```



```

</div>

{/* Product Details */}
<Card>
  <CardContent className="pt-6">
    <div className="grid grid-cols-2 gap-4">
      <div>
        <p className="text-sm text-muted-foreground">Category</p>
        <p className="font-medium">{product.category}</p>
      </div>
      <div>
        <p className="text-sm text-muted-foreground">Origin</p>
        <div className="flex items-center gap-1">
          <MapPin className="h-4 w-4" />
          <p className="font-medium">{product.origin || 'Local Farm'}</p>
        </div>
      </div>
      <div>
        <p className="text-sm text-muted-foreground">Harvest Date</p>
        <div className="flex items-center gap-1">
          <Calendar className="h-4 w-4" />
          <p className="font-medium">{product.harvestDate || 'This week'}</p>
        </div>
      </div>
      <div>
        <p className="text-sm text-muted-foreground">Farmer</p>
        <div className="flex items-center gap-1">
          <Users className="h-4 w-4" />
          <p className="font-medium">{product.farmer || 'Organic Farms Co.'}</p>
        </div>
      </div>
    </div>
  </CardContent>
</Card>
</div>

{/* Tabs for Additional Info */}
<div className="mt-12">
  <Tabs defaultValue="reviews" className="w-full">
    <TabsList className="grid w-full grid-cols-4">
      <TabsTrigger value="description">Description</TabsTrigger>
      <TabsTrigger value="specifications">Specifications</TabsTrigger>
      <TabsTrigger value="shipping">Shipping & Returns</TabsTrigger>
      <TabsTrigger value="reviews">Reviews</TabsTrigger>
    </TabsList>

    <TabsContent value="description" className="mt-6">

```

```
<div className="prose max-w-none">
  <h3>Product Details</h3>
  <p>{product.description}</p>
  <h3>Benefits</h3>
  <ul>
    <li>100% organic and pesticide-free</li>
    <li>Rich in essential vitamins and minerals</li>
    <li>Sustainably farmed with eco-friendly practices</li>
    <li>Supports local farmers and communities</li>
  </ul>
</div>
</TabsContent>

<TabsContent value="specifications">
  <div className="grid gap-4">
    <div className="grid grid-cols-2 gap-4">
      <div>
        <p className="text-sm text-muted-foreground">Weight</p>
        <p className="font-medium">1 kg</p>
      </div>
      <div>
        <p className="text-sm text-muted-foreground">Dimensions</p>
        <p className="font-medium">10 × 15 × 20 cm</p>
      </div>
      <div>
        <p className="text-sm text-muted-foreground">Storage</p>
        <p className="font-medium">Keep refrigerated</p>
      </div>
      <div>
        <p className="text-sm text-muted-foreground">Shelf Life</p>
        <p className="font-medium">7-10 days</p>
      </div>
    </div>
  </div>
</TabsContent>

<TabsContent value="shipping">
  <div className="space-y-4">
    <div>
      <h4 className="font-medium mb-2">Shipping Information</h4>
      <ul className="list-disc pl-5 space-y-1">
        <li>Free shipping on orders over €50</li>
        <li>Next-day delivery available in major cities</li>
        <li>Standard delivery: 2-3 business days</li>
        <li>Live tracking for all shipments</li>
      </ul>
    </div>
  </div>
</div>
```

```

        <h4 className="font-medium mb-2">Return Policy</h4>
        <p>30-day return policy for unused items in original packaging.</p>
      </div>
    </div>
  </TabsContent>

  <TabsContent value="reviews">
    <CommentSection productId={product.id} />
  </TabsContent>
</Tabs>
</div>

{/* Related Products */}
<div className="mt-12">
  <div className="flex items-center justify-between mb-6">
    <h2 className="text-2xl font-bold">You Might Also Like</h2>
    <Button variant="link">View All</Button>
  </div>
  <ProductSuggestions productId={product.id} categoryId={product.categoryId} />
</div>
</div>
</div>
);
}
...

```

## 📁 \*\*PAGES CUSTOMER\*\*

### ```src/app/(customer)/home/overview/page.tsx```

```tsx

'use client';

import { useState } from 'react';

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

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

import { Tabs, TabsContent, TabsList, TabsTrigger } from '@components/ui/tabs';

import { Badge } from '@components/ui/badge';

import {

Package,

ShoppingCart,

Truck,

Clock,

CheckCircle,

AlertCircle,

Star,

TrendingUp,

Calendar,

Download,  
Bell,  
Settings

```
} from 'lucide-react';  
import { useCustomerOverview } from '@/hooks/useCustomer';  
import { formatCurrency, formatDate } from '@/utils/formatters';  
import { Progress } from '@/components/ui/progress';  
import { RecentActivity } from '@/components/customer/RecentActivity';  
import { OrderStatusChart } from '@/components/customer/OrderStatusChart';  
import { QuickActions } from '@/components/customer/QuickActions';
```

```
export default function CustomerOverviewPage() {  
  const { overview, loading } = useCustomerOverview();  
  const [activeTab, setActiveTab] = useState('overview');
```

```
  const stats = [  
    {  
      title: 'Total Orders',  
      value: overview?.totalOrders || 0,  
      icon: ShoppingCart,  
      color: 'text-blue-600',  
      bgColor: 'bg-blue-50',  
      trend: '+12%',  
    },  
    {  
      title: 'Pending Orders',  
      value: overview?.pendingOrders || 0,  
      icon: Clock,  
      color: 'text-amber-600',  
      bgColor: 'bg-amber-50',  
      trend: '+3',  
    },  
    {  
      title: 'Delivered',  
      value: overview?.deliveredOrders || 0,  
      icon: CheckCircle,  
      color: 'text-green-600',  
      bgColor: 'bg-green-50',  
      trend: '+8',  
    },  
    {  
      title: 'Total Spent',  
      value: formatCurrency(overview?.totalSpent || 0),  
      icon: TrendingUp,  
      color: 'text-purple-600',  
      bgColor: 'bg-purple-50',  
      trend: '+15%',  
    },  
  ],
```

```
];
```

```
const recentOrders = [  
  {  
    id: 'ORD-7890',  
    product: 'Organic Vegetables Bundle',  
    date: '2024-01-15',  
    amount: 45.99,  
    status: 'delivered',  
    tracking: 'TRK-123456',  
  },  
  {  
    id: 'ORD-7889',  
    product: 'Fresh Fruits Assortment',  
    date: '2024-01-14',  
    amount: 32.50,  
    status: 'shipped',  
    tracking: 'TRK-123455',  
  },  
  {  
    id: 'ORD-7888',  
    product: 'Artisanal Bread Selection',  
    date: '2024-01-13',  
    amount: 28.75,  
    status: 'processing',  
    tracking: 'TRK-123454',  
  },  
  {  
    id: 'ORD-7887',  
    product: 'Dairy Products Package',  
    date: '2024-01-12',  
    amount: 52.30,  
    status: 'delivered',  
    tracking: 'TRK-123453',  
  },  
];
```

```
const statusColors = {  
  delivered: 'bg-green-100 text-green-800',  
  shipped: 'bg-blue-100 text-blue-800',  
  processing: 'bg-amber-100 text-amber-800',  
  pending: 'bg-gray-100 text-gray-800',  
  cancelled: 'bg-red-100 text-red-800',  
};
```

```
return (  
  <div className="space-y-6">
```

```

    <div className="flex flex-col lg:flex-row justify-between items-start lg:items-center
gap-4">
      <div>
        <h1 className="text-3xl font-bold tracking-tight">Welcome back, John!</h1>
        <p className="text-muted-foreground">
          Here's what's happening with your orders and account today.
        </p>
      </div>
      <div className="flex items-center gap-2">
        <Button variant="outline" size="sm">
          <Bell className="h-4 w-4 mr-2" />
          Notifications
        </Button>
        <Button variant="outline" size="sm">
          <Settings className="h-4 w-4 mr-2" />
          Settings
        </Button>
      </div>
    </div>

    { /* Quick Stats */ }
    <div className="grid gap-4 md:grid-cols-2 lg:grid-cols-4">
      {stats.map((stat, index) => (
        <Card key={index}>
          <CardHeader className="flex flex-row items-center justify-between space-y-0
pb-2">
            <CardTitle className="text-sm font-medium">
              {stat.title}
            </CardTitle>
            <div className={`p-2 rounded-full ${stat.bgColor}`}>
              <stat.icon className={`h-4 w-4 ${stat.color}`} />
            </div>
          </CardHeader>
          <CardContent>
            <div className="text-2xl font-bold">{stat.value}</div>
            <p className="text-xs text-muted-foreground">
              <span className="text-green-600">{stat.trend}</span> from last month
            </p>
          </CardContent>
        </Card>
      )))
    </div>

    <div className="grid gap-6 lg:grid-cols-3">
      { /* Left Column - Recent Orders & Activity */ }
      <div className="lg:col-span-2 space-y-6">
        <Tabs defaultValue="orders" className="w-full">
          <TabsList className="grid w-full grid-cols-3">

```

```

<TabsTrigger value="orders">Recent Orders</TabsTrigger>
<TabsTrigger value="tracking">Order Tracking</TabsTrigger>
<TabsTrigger value="activity">Recent Activity</TabsTrigger>
</TabsList>

<TabsContent value="orders" className="space-y-4">
  <Card>
    <CardHeader>
      <div className="flex items-center justify-between">
        <div>
          <CardTitle>Recent Orders</CardTitle>
          <CardDescription>
            Your latest purchases and their status
          </CardDescription>
        </div>
        <Button variant="ghost" size="sm">
          View All
        </Button>
      </div>
    </CardHeader>
    <CardContent>
      <div className="space-y-4">
        {recentOrders.map((order) => (
          <div
            key={order.id}
            className="flex items-center justify-between p-4 border rounded-lg
            hover:bg-gray-50 dark:hover:bg-gray-800 transition-colors"
          >
            <div className="flex items-center gap-4">
              <div className="bg-blue-50 p-2 rounded-lg">
                <Package className="h-6 w-6 text-blue-600" />
              </div>
              <div>
                <p className="font-medium">{order.product}</p>
                <div className="flex items-center gap-4 text-sm text-muted-foreground">
                  <span>Order #{order.id}</span>
                  <span>•</span>
                  <span>{formatDate(order.date)}</span>
                  <span>•</span>
                  <span>{formatCurrency(order.amount)}</span>
                </div>
              </div>
            </div>
            <div className="flex items-center gap-3">
              <Badge className={statusColors[order.status as keyof typeof
statusColors]}>
                {order.status}
              </Badge>
            </div>
          </div>
        )

```

```

        <Button variant="ghost" size="sm">
          Track
        </Button>
      </div>
    </div>
  )}}
</div>
</CardContent>
</Card>
</TabsContent>

<TabsContent value="tracking">
  <Card>
    <CardHeader>
      <CardTitle>Order Tracking</CardTitle>
      <CardDescription>
        Track your shipments in real-time
      </CardDescription>
    </CardHeader>
    <CardContent>
      <div className="space-y-6">
        {recentOrders
          .filter(order => order.status === 'shipped')
          .map((order) => (
            <div key={order.id} className="space-y-4">
              <div className="flex items-center justify-between">
                <div>
                  <p className="font-medium">Order #{order.id}</p>
                  <p className="text-sm text-muted-foreground">
                    Tracking: {order.tracking}
                  </p>
                </div>
                <Badge variant="outline">In Transit</Badge>
              </div>
              <div className="space-y-2">
                <div className="flex justify-between text-sm">
                  <span>Order Placed</span>
                  <span>Processing</span>
                  <span>Shipped</span>
                  <span>Out for Delivery</span>
                  <span>Delivered</span>
                </div>
                <Progress value={60} className="h-2" />
                <p className="text-sm text-muted-foreground">
                  Estimated delivery: Tomorrow, 2-4 PM
                </p>
              </div>
            </div>
          ))}
      </div>
    </CardContent>
  </Card>
</TabsContent>

```



```

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

<TabsContent value="activity">
  <RecentActivity />
</TabsContent>
</Tabs>
</div>

```

```

{/* Right Column - Quick Actions & Stats */}
<div className="space-y-6">
  <QuickActions />

```

```

  <Card>
    <CardHeader>
      <CardTitle>Order Status</CardTitle>
      <CardDescription>
        Distribution of your orders by status
      </CardDescription>
    </CardHeader>
    <CardContent>
      <OrderStatusChart />
    </CardContent>
  </Card>

```

```

  <Card>
    <CardHeader>
      <CardTitle>Loyalty Program</CardTitle>
      <CardDescription>
        Your rewards and benefits
      </CardDescription>
    </CardHeader>
    <CardContent>
      <div className="space-y-4">
        <div className="flex items-center justify-between">
          <div className="flex items-center gap-2">
            <Star className="h-5 w-5 text-amber-500" />
            <span className="font-medium">Gold Member</span>
          </div>
          <Badge variant="secondary">Level 3</Badge>
        </div>
        <div className="space-y-2">
          <div className="flex justify-between text-sm">
            <span>Progress to Platinum</span>
            <span>75%</span>
          </div>

```

```

    </div>
    <Progress value={75} className="h-2" />
  </div>
  <div className="grid grid-cols-2 gap-4 text-sm">
    <div>
      <p className="text-muted-foreground">Points</p>
      <p className="font-medium">1,250</p>
    </div>
    <div>
      <p className="text-muted-foreground">Next Reward</p>
      <p className="font-medium">Free Delivery</p>
    </div>
  </div>
</CardContent>
</Card>
</div>
</div>

```

```

{/* Recommendations */}
<Card>
  <CardHeader>
    <div className="flex items-center justify-between">
      <div>
        <CardTitle>Recommended For You</CardTitle>
        <CardDescription>
          Products you might like based on your purchase history
        </CardDescription>
      </div>
      <Button variant="ghost" size="sm">
        Browse All
      </Button>
    </div>
  </CardHeader>
  <CardContent>
    <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-4 gap-4">
      {[
        { name: 'Seasonal Fruit Basket', price: 29.99, category: 'Fruits' },
        { name: 'Organic Vegetable Box', price: 34.99, category: 'Vegetables' },
        { name: 'Artisanal Cheese Selection', price: 42.50, category: 'Dairy' },
        { name: 'Fresh Bread Assortment', price: 18.75, category: 'Bakery' },
      ]}.map((product, index) => (
        <div key={index} className="border rounded-lg p-4 hover:shadow-md transition-shadow">
          <div className="aspect-square bg-gray-100 rounded-md mb-3"></div>
          <div>
            <p className="font-medium">{product.name}</p>
            <p className="text-sm text-muted-foreground">{product.category}</p>

```

```

        <div className="flex items-center justify-between mt-2">
          <span className="font-bold text-blue-600">
            {formatCurrency(product.price)}
          </span>
          <Button size="sm">Add</Button>
        </div>
      </div>
    </div>
  )))
</div>
</CardContent>
</Card>
</div>
);
}
...

```

## 📁 **\*\*PAGES CHAT\*\***

### **\*\*`src/app/(chat)/page.tsx`\*\***

**```tsx**

'use client';

```

import { useState, useEffect, useRef } from 'react';
import { Card, CardContent, CardHeader, CardTitle } from '@components/ui/card';
import { Button } from '@components/ui/button';
import { Input } from '@components/ui/input';
import { ScrollArea } from '@components/ui/scroll-area';
import { Badge } from '@components/ui/badge';
import { Tabs, TabsContent, TabsList, TabsTrigger } from '@components/ui/tabs';
import {
  Search,
  Users,
  MessageSquare,
  Phone,
  Video,
  MoreVertical,
  Paperclip,
  Smile,
  Send,
  Bell,
  Check,
  CheckCheck,
  Clock
} from 'lucide-react';
import { useChatStore } from '@stores/useChatStore';
import { formatDate, formatTime } from '@utils/formatters';
import { Avatar, AvatarFallback, AvatarImage } from '@components/ui/avatar';

```

```

import { Separator } from '@components/ui/separator';

interface ChatMessage {
  id: string;
  sender: {
    id: string;
    name: string;
    avatar?: string;
    role: string;
  };
  content: string;
  timestamp: Date;
  isRead: boolean;
  type: 'text' | 'image' | 'file';
}

interface ChatRoom {
  id: string;
  name: string;
  type: 'private' | 'group';
  participants: Array<{
    id: string;
    name: string;
    avatar?: string;
    role: string;
  }>;
  lastMessage: {
    content: string;
    timestamp: Date;
    sender: string;
  };
  unreadCount: number;
  isOnline: boolean;
}

export default function ChatPage() {
  const [activeRoom, setActiveRoom] = useState<string | null>(null);
  const [message, setMessage] = useState("");
  const [searchQuery, setSearchQuery] = useState("");
  const messagesEndRef = useRef<HTMLDivElement>(null);

  const { rooms, messages, sendMessage, markAsRead, connect, disconnect } =
    useChatStore();

  useEffect(() => {
    connect();
    return () => disconnect();
  }, []);

```

```

useEffect(() => {
  messagesEndRef.current?.scrollIntoView({ behavior: 'smooth' });
}, [messages]);

const filteredRooms = rooms.filter(room =>
  room.name.toLowerCase().includes(searchQuery.toLowerCase()) ||
  room.participants.some(p =>
    p.name.toLowerCase().includes(searchQuery.toLowerCase())
  ));

const activeRoomData = rooms.find(room => room.id === activeRoom);
const roomMessages = activeRoom ? messages[activeRoom] || [] : [];

const handleSendMessage = () => {
  if (message.trim() && activeRoom) {
    sendMessage(activeRoom, message);
    setMessage("");
  }
};

const handleKeyPress = (e: React.KeyboardEvent) => {
  if (e.key === 'Enter' && !e.shiftKey) {
    e.preventDefault();
    handleSendMessage();
  }
};

return (
  <div className="flex h-[calc(100vh-4rem)] bg-gray-50 dark:bg-gray-900">
    { /* Sidebar - Chat List */ }
    <div className="w-full md:w-80 border-r bg-white dark:bg-gray-800">
      <div className="p-4 border-b">
        <div className="flex items-center justify-between mb-4">
          <h2 className="text-xl font-bold">Messages</h2>
          <Button size="icon" variant="ghost">
            <Bell className="h-5 w-5" />
          </Button>
        </div>
        <div className="relative">
          <Search className="absolute left-3 top-3 h-4 w-4 text-gray-400" />
          <Input
            placeholder="Search conversations..."
            value={searchQuery}
            onChange={(e) => setSearchQuery(e.target.value)}
            className="pl-10"
          />
        </div>
      </div>
    </div>
  </div>

```

```

</div>

<Tabs defaultValue="all" className="w-full">
  <div className="px-4">
    <TabsList className="grid w-full grid-cols-3">
      <TabsTrigger value="all">All</TabsTrigger>
      <TabsTrigger value="unread">Unread</TabsTrigger>
      <TabsTrigger value="groups">Groups</TabsTrigger>
    </TabsList>
  </div>

  <ScrollArea className="h-[calc(100vh-12rem)]">
    <TabsContent value="all" className="mt-0">
      <div className="p-2">
        {filteredRooms.map((room) => (
          <div
            key={room.id}
            className={`flex items-center gap-3 p-3 rounded-lg cursor-pointer
transition-colors ${
              activeRoom === room.id
                ? 'bg-blue-50 dark:bg-blue-900/20'
                : 'hover:bg-gray-50 dark:hover:bg-gray-700'
            }}
            onClick={() => {
              setActiveRoom(room.id);
              markAsRead(room.id);
            }}
          >
            <div className="relative">
              <Avatar>
                <AvatarImage src={room.participants[0]?.avatar} />
                <AvatarFallback>
                  {room.name.charAt(0).toUpperCase()}
                </AvatarFallback>
              </Avatar>
              {room.isOnline && (
                <div className="absolute bottom-0 right-0 h-3 w-3 bg-green-500
rounded-full border-2 border-white dark:border-gray-800" />
              )}
            </div>
            <div className="flex-1 min-w-0">
              <div className="flex items-center justify-between">
                <p className="font-medium truncate">{room.name}</p>
                <span className="text-xs text-muted-foreground">
                  {formatTime(room.lastMessage.timestamp)}
                </span>
              </div>
            </div>
          </div>
        )}
      </div>
    </TabsContent>
  </ScrollArea>
</Tabs>

```

```

        <p className="text-sm text-muted-foreground truncate">
            {room.lastMessage.sender}: {room.lastMessage.content}
        </p>
        {room.unreadCount > 0 && (
            <Badge className="ml-2 bg-blue-600">
                {room.unreadCount}
            </Badge>
        )}
    </div>
</div>
</div>
    )}}
</div>
</TabsContent>
</ScrollArea>
</Tabs>
</div>

{/* Main Chat Area */}
<div className="flex-1 flex flex-col">
    {activeRoomData ? (
        <>
            {/* Chat Header */}
            <div className="p-4 border-b bg-white dark:bg-gray-800">
                <div className="flex items-center justify-between">
                    <div className="flex items-center gap-3">
                        <Avatar>
                            <AvatarImage src={activeRoomData.participants[0]?.avatar} />
                            <AvatarFallback>
                                {activeRoomData.name.charAt(0).toUpperCase()}
                            </AvatarFallback>
                        </Avatar>
                        <div>
                            <h3 className="font-bold">{activeRoomData.name}</h3>
                            <div className="flex items-center gap-2 text-sm text-muted-foreground">
                                {activeRoomData.type === 'group' && (
                                    <>
                                        <Users className="h-3 w-3" />
                                        <span>{activeRoomData.participants.length} members</span>
                                    </>
                                )}
                                {activeRoomData.isOnline && (
                                    <>
                                        <div className="h-2 w-2 bg-green-500 rounded-full"></div>
                                        <span>Online</span>
                                    </>
                                )}
                            </div>
                        </div>
                    </div>
                </div>
            </div>

```

```

    </div>
  </div>
  <div className="flex items-center gap-2">
    <Button size="icon" variant="ghost">
      <Phone className="h-5 w-5" />
    </Button>
    <Button size="icon" variant="ghost">
      <Video className="h-5 w-5" />
    </Button>
    <Button size="icon" variant="ghost">
      <MoreVertical className="h-5 w-5" />
    </Button>
  </div>
</div>
</div>
</div>

{/* Messages Area */}
<ScrollArea className="flex-1 p-4">
  <div className="space-y-4">
    {roomMessages.map((msg, index) => {
      const isUser = msg.sender.id === 'current-user';
      const showAvatar = index === 0 || roomMessages[index - 1]?.sender.id !==
msg.sender.id;
      const showDate = index === 0 ||
        new Date(msg.timestamp).toDateString() !==
        new Date(roomMessages[index - 1].timestamp).toDateString();

      return (
        <div key={msg.id}>
          {showDate && (
            <div className="flex justify-center my-4">
              <Badge variant="secondary" className="px-3">
                {formatDate(msg.timestamp)}
              </Badge>
            </div>
          )}
          <div
            className={`flex gap-3 ${isUser ? 'flex-row-reverse' : ''}`}
          >
            {showAvatar && !isUser && (
              <Avatar className="h-8 w-8">
                <AvatarImage src={msg.sender.avatar} />
                <AvatarFallback>{msg.sender.name.charAt(0)}</AvatarFallback>
              </Avatar>
            )}
            {!isUser && !showAvatar && <div className="w-8" />}
            <div className={`max-w-[70%] ${isUser ? 'text-right' : ''}`}>
              {showAvatar && (

```



```

        <p className={`text-sm font-medium mb-1 ${isUser ? 'text-right' : ''}`}>
            {msg.sender.name}
        </p>
    )}
    <div
        className={`rounded-2xl px-4 py-2 ${
            isUser
                ? 'bg-blue-600 text-white rounded-br-none'
                : 'bg-gray-100 dark:bg-gray-700 rounded-bl-none'
        }`}
    >
        <p>{msg.content}</p>
    </div>
    <div className={`flex items-center gap-1 mt-1 text-xs
text-muted-foreground ${isUser ? 'justify-end' : ''}`}>
        <span>{formatTime(msg.timestamp)}</span>
        {isUser && (
            <>
                {msg.isRead ? (
                    <CheckCheck className="h-3 w-3" />
                ) : (
                    <Check className="h-3 w-3" />
                )}
            </>
        )}
    </div>
</div>
</div>
</div>
);
}}
<div ref={messagesEndRef} />
</div>
</ScrollArea>

```

```

{/* Message Input */}
<div className="p-4 border-t bg-white dark:bg-gray-800">
    <div className="flex items-end gap-2">
        <Button size="icon" variant="ghost">
            <Paperclip className="h-5 w-5" />
        </Button>
        <div className="flex-1 relative">
            <Input
                placeholder="Type your message..."
                value={message}
                onChange={(e) => setMessage(e.target.value)}
                onKeyPress={handleKeyPress}
                className="pr-12"
            />
        </div>
    </div>

```

```

        multiline
        rows={1}
      />
      <Button
        size="icon"
        variant="ghost"
        className="absolute right-2 top-1/2 transform -translate-y-1/2"
      >
        <Smile className="h-5 w-5" />
      </Button>
    </div>
    <Button
      size="icon"
      onClick={handleSendMessage}
      disabled={!message.trim()}
      className="bg-blue-600 hover:bg-blue-700"
    >
      <Send className="h-5 w-5" />
    </Button>
  </div>
</div>
</>
): (
  /* Empty State */
  <div className="flex-1 flex flex-col items-center justify-center p-8">
    <div className="w-24 h-24 bg-gray-100 dark:bg-gray-700 rounded-full flex
items-center justify-center mb-6">
      <MessageSquare className="h-12 w-12 text-gray-400" />
    </div>
    <h3 className="text-xl font-bold mb-2">Welcome to AgroDeep Chat</h3>
    <p className="text-muted-foreground text-center max-w-md mb-6">
      Select a conversation from the sidebar to start chatting with farmers,
      suppliers, or customer support.
    </p>
    <Button>
      <Users className="h-4 w-4 mr-2" />
      Start New Conversation
    </Button>
  </div>
  )}
</div>
</div>
);
}
...

```

## 📁 \*\*PAGES NOTIFICATIONS\*\*

```

### **`src/app/(notifications)/page.tsx`**
``tsx
'use client';

import { useState } from 'react';
import { Card, CardContent, CardHeader, CardTitle } from '@components/ui/card';
import { Button } from '@components/ui/button';
import { Badge } from '@components/ui/badge';
import { Separator } from '@components/ui/separator';
import { Tabs, TabsContent, TabsList, TabsTrigger } from '@components/ui/tabs';
import {
  Bell,
  Check,
  Trash2,
  Settings,
  Filter,
  AlertTriangle,
  ShoppingCart,
  Package,
  DollarSign,
  MessageSquare,
  Calendar,
  Users
} from 'lucide-react';
import { useNotificationStore } from '@stores/useNotificationStore';
import { formatDate, formatTime } from '@utils/formatters';
import { Switch } from '@components/ui/switch';
import { Label } from '@components/ui/label';

type NotificationType = 'all' | 'unread' | 'system' | 'orders' | 'promotions';

export default function NotificationsPage() {
  const [activeTab, setActiveTab] = useState<NotificationType>('all');
  const [selectedNotifications, setSelectedNotifications] = useState<string[]>([]);
  const { notifications, markAsRead, markAllAsRead, deleteNotification,
notificationPreferences, updatePreferences } = useNotificationStore();

  const getNotificationIcon = (type: string) => {
    switch (type) {
      case 'order':
        return ShoppingCart;
      case 'system':
        return Bell;
      case 'promotion':
        return DollarSign;
      case 'chat':
        return MessageSquare;
      case 'event':

```

```
    return Calendar;
  case 'alert':
    return AlertTriangle;
  default:
    return Bell;
}
};
```

```
const getNotificationColor = (type: string) => {
  switch (type) {
    case 'order':
      return 'bg-blue-100 text-blue-600';
    case 'system':
      return 'bg-gray-100 text-gray-600';
    case 'promotion':
      return 'bg-green-100 text-green-600';
    case 'chat':
      return 'bg-purple-100 text-purple-600';
    case 'alert':
      return 'bg-red-100 text-red-600';
    default:
      return 'bg-gray-100 text-gray-600';
  }
};
```

```
const filteredNotifications = notifications.filter(notification => {
  if (activeTab === 'all') return true;
  if (activeTab === 'unread') return !notification.isRead;
  return notification.type === activeTab;
});
```

```
const unreadCount = notifications.filter(n => !n.isRead).length;
```

```
const handleSelectAll = () => {
  if (selectedNotifications.length === filteredNotifications.length) {
    setSelectedNotifications([]);
  } else {
    setSelectedNotifications(filteredNotifications.map(n => n.id));
  }
};
```

```
const handleBulkAction = (action: 'read' | 'delete') => {
  selectedNotifications.forEach(id => {
    if (action === 'read') {
      markAsRead(id);
    } else {
      deleteNotification(id);
    }
  })
};
```

```

});
setSelectedNotifications([]);
};

return (
  <div className="space-y-6">
    <div className="flex flex-col lg:flex-row justify-between items-start lg:items-center gap-4">
      <div>
        <h1 className="text-3xl font-bold tracking-tight">Notifications</h1>
        <p className="text-muted-foreground">
          Stay updated with your orders, messages, and platform alerts
        </p>
      </div>
      <div className="flex items-center gap-2">
        {unreadCount > 0 && (
          <Button variant="outline" onClick={markAllAsRead}>
            <Check className="h-4 w-4 mr-2" />
            Mark all as read
          </Button>
        )}
        <Button variant="outline">
          <Settings className="h-4 w-4 mr-2" />
          Settings
        </Button>
        <Button variant="outline">
          <Filter className="h-4 w-4 mr-2" />
          Filter
        </Button>
      </div>
    </div>

    { /* Stats */}
    <div className="grid gap-4 md:grid-cols-2 lg:grid-cols-4">
      <Card>
        <CardHeader className="flex flex-row items-center justify-between space-y-0 pb-2">
          <CardTitle className="text-sm font-medium">Total</CardTitle>
          <Bell className="h-4 w-4 text-muted-foreground" />
        </CardHeader>
        <CardContent>
          <div className="text-2xl font-bold">{notifications.length}</div>
          <p className="text-xs text-muted-foreground">All notifications</p>
        </CardContent>
      </Card>
      <Card>
        <CardHeader className="flex flex-row items-center justify-between space-y-0 pb-2">
          <CardTitle className="text-sm font-medium">Unread</CardTitle>
          <AlertTriangle className="h-4 w-4 text-muted-foreground" />

```

```

</CardHeader>
<CardContent>
  <div className="text-2xl font-bold">{unreadCount}</div>
  <p className="text-xs text-muted-foreground">Require attention</p>
</CardContent>
</Card>
<Card>
  <CardHeader className="flex flex-row items-center justify-between space-y-0 pb-2">
    <CardTitle className="text-sm font-medium">Orders</CardTitle>
    <ShoppingCart className="h-4 w-4 text-muted-foreground" />
  </CardHeader>
  <CardContent>
    <div className="text-2xl font-bold">
      {notifications.filter(n => n.type === 'order').length}
    </div>
    <p className="text-xs text-muted-foreground">Order updates</p>
  </CardContent>
</Card>
<Card>
  <CardHeader className="flex flex-row items-center justify-between space-y-0 pb-2">
    <CardTitle className="text-sm font-medium">Today</CardTitle>
    <Calendar className="h-4 w-4 text-muted-foreground" />
  </CardHeader>
  <CardContent>
    <div className="text-2xl font-bold">
      {notifications.filter(n =>
        new Date(n.timestamp).toDateString() === new Date().toDateString()
      ).length}
    </div>
    <p className="text-xs text-muted-foreground">Notifications today</p>
  </CardContent>
</Card>
</div>

<div className="grid gap-6 lg:grid-cols-3">
  {/* Left Column - Notifications List */}
  <div className="lg:col-span-2">
    <Card>
      <CardHeader>
        <div className="flex flex-col sm:flex-row sm:items-center justify-between gap-4">
          <CardTitle>Your Notifications</CardTitle>
          {selectedNotifications.length > 0 && (
            <div className="flex items-center gap-2">
              <Button
                variant="outline"
                size="sm"
                onClick={() => handleBulkAction('read')}
              >

```

```

        <Check className="h-4 w-4 mr-2" />
        Mark as read
    </Button>
    <Button
        variant="outline"
        size="sm"
        onClick={() => handleBulkAction('delete')}
    >
        <Trash2 className="h-4 w-4 mr-2" />
        Delete
    </Button>
</div>
))
</div>
</CardHeader>
<CardContent>
    <Tabs value={activeTab} onValueChange={(v) => setActiveTab(v as
NotificationType)}>
        <TabsList className="grid w-full grid-cols-5">
            <TabsTrigger value="all">All</TabsTrigger>
            <TabsTrigger value="unread">
                Unread {unreadCount > 0 && `${unreadCount}`}
            </TabsTrigger>
            <TabsTrigger value="system">System</TabsTrigger>
            <TabsTrigger value="orders">Orders</TabsTrigger>
            <TabsTrigger value="promotions">Promotions</TabsTrigger>
        </TabsList>

        <div className="mt-4">
            <div className="flex items-center justify-between mb-4">
                <Button
                    variant="ghost"
                    size="sm"
                    onClick={handleSelectAll}
                >
                    {selectedNotifications.length === filteredNotifications.length
                        ? 'Deselect all'
                        : 'Select all'}
                </Button>
                <span className="text-sm text-muted-foreground">
                    {filteredNotifications.length} notifications
                </span>
            </div>

            <div className="space-y-2">
                {filteredNotifications.length === 0 ? (
                    <div className="text-center py-12">
                        <Bell className="h-12 w-12 text-gray-300 mx-auto mb-4" />

```

```

    <h3 className="text-lg font-medium">No notifications</h3>
    <p className="text-muted-foreground">
      You're all caught up!
    </p>
  </div>
) : (
  filteredNotifications.map((notification) => {
    const Icon = getNotificationIcon(notification.type);
    return (
      <div
        key={notification.id}
        className={`p-4 rounded-lg border transition-colors ${
          notification.isRead
            ? 'bg-white dark:bg-gray-800'
            : 'bg-blue-50 dark:bg-blue-900/20 border-blue-200 dark:border-blue-800'
        } ${
          selectedNotifications.includes(notification.id)
            ? 'ring-2 ring-blue-500'
            : ''
        }`}
        onClick={() => {
          if (!notification.isRead) {
            markAsRead(notification.id);
          }
        }}
      >
        <div className="flex items-start gap-4">
          <div
            className={`p-2 rounded-full ${getNotificationColor(
              notification.type
            )}`}
          >
            <Icon className="h-4 w-4" />
          </div>
          <div className="flex-1">
            <div className="flex items-start justify-between">
              <div>
                <h4 className="font-medium">
                  {notification.title}
                </h4>
                <p className="text-sm text-muted-foreground mt-1">
                  {notification.message}
                </p>
              </div>
              <div className="flex items-center gap-2">
                <span className="text-xs text-muted-foreground">
                  {formatTime(notification.timestamp)}
                </span>
              </div>
            </div>
          </div>
        </div>
      </div>
    );
  })
);

```



```

<input
  type="checkbox"
  checked={selectedNotifications.includes(
    notification.id
  )}
  onChange={(e) => {
    if (e.target.checked) {
      setSelectedNotifications([
        ...selectedNotifications,
        notification.id,
      ]);
    } else {
      setSelectedNotifications(
        selectedNotifications.filter(
          (id) => id !== notification.id
        )
      );
    }
  }}
  onClick={(e) => e.stopPropagation()}
  className="h-4 w-4"
/>
</div>
</div>
{notification.metadata && (
  <div className="mt-2">
    {notification.metadata.orderId && (
      <Badge variant="outline" className="mr-2">
        Order #{notification.metadata.orderId}
      </Badge>
    )}
    {notification.metadata.product && (
      <Badge variant="secondary" className="mr-2">
        {notification.metadata.product}
      </Badge>
    )}
  </div>
)}
</div>
</div>
</div>
);
})
}}
</div>
</div>
</CardContent>

```

```
</Card>
</div>
```

```
{/* Right Column - Settings */}
<div className="space-y-6">
  <Card>
    <CardHeader>
      <CardTitle>Notification Settings</CardTitle>
    </CardHeader>
    <CardContent>
      <div className="space-y-4">
        <div className="flex items-center justify-between">
          <div className="space-y-0.5">
            <Label>Email Notifications</Label>
            <p className="text-sm text-muted-foreground">
              Receive notifications via email
            </p>
          </div>
          <Switch
            checked={notificationPreferences.email}
            onCheckedChange={(checked) =>
              updatePreferences({ email: checked })
            }
          />
        </div>
        <Separator />
        <div className="flex items-center justify-between">
          <div className="space-y-0.5">
            <Label>Push Notifications</Label>
            <p className="text-sm text-muted-foreground">
              Browser push notifications
            </p>
          </div>
          <Switch
            checked={notificationPreferences.push}
            onCheckedChange={(checked) =>
              updatePreferences({ push: checked })
            }
          />
        </div>
        <Separator />
        <div className="flex items-center justify-between">
          <div className="space-y-0.5">
            <Label>SMS Notifications</Label>
            <p className="text-sm text-muted-foreground">
              Text message alerts
            </p>
          </div>
        </div>
      </div>
    </CardContent>
  </Card>
</div>
```

```

        <Switch
          checked={notificationPreferences.sms}
          onChange={({checked}) =>
            updatePreferences({ sms: checked })
          }
        />
      </div>
    </div>
  </CardContent>
</Card>

<Card>
  <CardHeader>
    <CardTitle>Notification Types</CardTitle>
  </CardHeader>
  <CardContent>
    <div className="space-y-4">
      <div className="flex items-center justify-between">
        <div className="space-y-0.5">
          <Label>Order Updates</Label>
          <p className="text-sm text-muted-foreground">
            Order status changes
          </p>
        </div>
        <Switch defaultChecked />
      </div>
      <div className="flex items-center justify-between">
        <div className="space-y-0.5">
          <Label>Promotions</Label>
          <p className="text-sm text-muted-foreground">
            Special offers and discounts
          </p>
        </div>
        <Switch defaultChecked />
      </div>
      <div className="flex items-center justify-between">
        <div className="space-y-0.5">
          <Label>System Alerts</Label>
          <p className="text-sm text-muted-foreground">
            Platform updates and maintenance
          </p>
        </div>
        <Switch defaultChecked />
      </div>
      <div className="flex items-center justify-between">
        <div className="space-y-0.5">
          <Label>Chat Messages</Label>
          <p className="text-sm text-muted-foreground">

```

```

        New messages from chats
      </p>
    </div>
    <Switch defaultChecked />
  </div>
</div>
</CardContent>
</Card>
</div>
</div>
</div>
);
}
...

```

## 📁 \*\*PAGES REPORTS\*\*

```

### **`src/app/(reports)/page.tsx`**
```tsx
'use client';

```

```

import { useState } from 'react';
import { Card, CardContent, CardDescription, CardHeader, CardTitle } from
'@/components/ui/card';
import { Button } from '@/components/ui/button';
import { Tabs, TabsContent, TabsList, TabsTrigger } from '@/components/ui/tabs';
import { Select, SelectContent, SelectItem, SelectTrigger, SelectValue } from
'@/components/ui/select';
import { DateRangePicker } from '@/components/ui/date-range-picker';
import {
  Download,
  Printer,
  Filter,
  BarChart3,
  PieChart,
  TrendingUp,
  FileText,
  Calendar,
  RefreshCw,
  Eye,
  Share2,
  Mail
} from 'lucide-react';
import { formatCurrency, formatDate } from '@/utils/formatters';
import {
  BarChart,
  Bar,
  XAxis,

```

```

YAxis,
CartesianGrid,
Tooltip,
Legend,
ResponsiveContainer,
PieChart as RechartsPie,
Pie,
Cell
} from 'recharts';

const COLORS = ['#0088FE', '#00C49F', '#FFBB28', '#FF8042', '#8884D8'];

export default function ReportsPage() {
  const [dateRange, setDateRange] = useState({
    from: new Date(new Date().setDate(new Date().getDate() - 30)),
    to: new Date(),
  });
  const [reportType, setReportType] = useState('sales');
  const [format, setFormat] = useState('pdf');

  const salesData = [
    { month: 'Jan', revenue: 42000, orders: 150, customers: 120 },
    { month: 'Feb', revenue: 52000, orders: 180, customers: 145 },
    { month: 'Mar', revenue: 61000, orders: 210, customers: 165 },
    { month: 'Apr', revenue: 78000, orders: 250, customers: 190 },
    { month: 'May', revenue: 85000, orders: 280, customers: 210 },
    { month: 'Jun', revenue: 92000, orders: 310, customers: 230 },
  ];

  const categoryData = [
    { name: 'Vegetables', value: 35 },
    { name: 'Fruits', value: 25 },
    { name: 'Grains', value: 20 },
    { name: 'Dairy', value: 15 },
    { name: 'Meat', value: 5 },
  ];

  const topProducts = [
    { name: 'Organic Tomatoes', revenue: 18750, units: 1250 },
    { name: 'Fresh Apples', revenue: 14700, units: 980 },
    { name: 'Whole Wheat', revenue: 11250, units: 750 },
    { name: 'Fresh Milk', revenue: 9300, units: 620 },
    { name: 'Free-range Eggs', revenue: 8100, units: 540 },
  ];

  const reports = [
    {
      id: 1,

```

```

    name: 'Monthly Sales Report',
    type: 'sales',
    period: 'January 2024',
    generated: '2024-01-31',
    size: '2.4 MB',
    status: 'ready',
  },
  {
    id: 2,
    name: 'Inventory Analysis',
    type: 'inventory',
    period: 'Q4 2023',
    generated: '2024-01-15',
    size: '1.8 MB',
    status: 'ready',
  },
  {
    id: 3,
    name: 'Customer Insights',
    type: 'customers',
    period: 'December 2023',
    generated: '2024-01-10',
    size: '3.2 MB',
    status: 'ready',
  },
  {
    id: 4,
    name: 'Annual Summary',
    type: 'annual',
    period: '2023',
    generated: '2024-01-05',
    size: '5.1 MB',
    status: 'ready',
  },
];

```

```

const handleGenerateReport = () => {
  // Generate report logic
};

```

```

const handleExport = () => {
  // Export logic
};

```

```

return (
  <div className="space-y-6">
    <div className="flex flex-col lg:flex-row justify-between items-start lg:items-center gap-4">

```

```

<div>
  <h1 className="text-3xl font-bold tracking-tight">Reports & Analytics</h1>
  <p className="text-muted-foreground">
    Generate and analyze detailed reports for better decision making
  </p>
</div>
<div className="flex items-center gap-2">
  <Button variant="outline">
    <RefreshCw className="h-4 w-4 mr-2" />
    Refresh
  </Button>
  <Button onClick={handleGenerateReport}>
    <FileText className="h-4 w-4 mr-2" />
    Generate Report
  </Button>
</div>
</div>

```

```

{/* Report Configuration */}
<Card>
  <CardHeader>
    <CardTitle>Report Configuration</CardTitle>
    <CardDescription>
      Configure the parameters for your report
    </CardDescription>
  </CardHeader>
  <CardContent>
    <div className="grid gap-4 md:grid-cols-2 lg:grid-cols-4">
      <div className="space-y-2">
        <label className="text-sm font-medium">Report Type</label>
        <Select value={reportType} onChange={setReportType}>
          <SelectTrigger>
            <SelectValue placeholder="Select report type" />
          </SelectTrigger>
          <SelectContent>
            <SelectItem value="sales">Sales Report</SelectItem>
            <SelectItem value="inventory">Inventory Report</SelectItem>
            <SelectItem value="customers">Customer Report</SelectItem>
            <SelectItem value="products">Product Report</SelectItem>
            <SelectItem value="financial">Financial Report</SelectItem>
          </SelectContent>
        </Select>
      </div>
      <div className="space-y-2">
        <label className="text-sm font-medium">Date Range</label>
        <DateRangePicker
          date={dateRange}
          onChange={setDateRange}

```

```

    />
  </div>
  <div className="space-y-2">
    <label className="text-sm font-medium">Format</label>
    <Select value={format} onChange={setFormat}>
      <SelectTrigger>
        <SelectValue placeholder="Select format" />
      </SelectTrigger>
      <SelectContent>
        <SelectItem value="pdf">PDF Document</SelectItem>
        <SelectItem value="excel">Excel Spreadsheet</SelectItem>
        <SelectItem value="csv">CSV File</SelectItem>
        <SelectItem value="json">JSON Data</SelectItem>
      </SelectContent>
    </Select>
  </div>
  <div className="space-y-2">
    <label className="text-sm font-medium">Actions</label>
    <div className="flex gap-2">
      <Button variant="outline" className="flex-1">
        <Filter className="h-4 w-4 mr-2" />
        Filters
      </Button>
      <Button onClick={handleExport} className="flex-1">
        <Download className="h-4 w-4 mr-2" />
        Export
      </Button>
    </div>
  </div>
</CardContent>
</Card>

<Tabs defaultValue="overview" className="space-y-4">
  <TabsList>
    <TabsTrigger value="overview">Overview</TabsTrigger>
    <TabsTrigger value="sales">Sales Analysis</TabsTrigger>
    <TabsTrigger value="inventory">Inventory</TabsTrigger>
    <TabsTrigger value="customers">Customers</TabsTrigger>
  </TabsList>

  <TabsContent value="overview" className="space-y-4">
    <div className="grid gap-4 md:grid-cols-2 lg:grid-cols-3">
      <Card className="col-span-2">
        <CardHeader>
          <CardTitle>Sales Performance</CardTitle>
          <CardDescription>
            Revenue and order trends over time

```



```

    </CardDescription>
  </CardHeader>
  <CardContent>
    <div className="h-[300px]">
      <ResponsiveContainer width="100%" height="100%">
        <BarChart data={salesData}>
          <CartesianGrid strokeDasharray="3 3" />
          <XAxis dataKey="month" />
          <YAxis />
          <Tooltip
            formatter={(value) => [
              reportType === 'sales'
                ? formatCurrency(value as number)
                : value,
              reportType === 'sales' ? 'Revenue' : 'Orders'
            ]}
          />
          <Legend />
          <Bar dataKey="revenue" fill="#2563eb" name="Revenue" />
          <Bar dataKey="orders" fill="#10b981" name="Orders" />
        </BarChart>
      </ResponsiveContainer>
    </div>
  </CardContent>
</Card>

<Card>
  <CardHeader>
    <CardTitle>Revenue by Category</CardTitle>
    <CardDescription>
      Distribution across product categories
    </CardDescription>
  </CardHeader>
  <CardContent>
    <div className="h-[300px]">
      <ResponsiveContainer width="100%" height="100%">
        <RechartsPie>
          <Pie
            data={categoryData}
            cx="50%"
            cy="50%"
            labelLine={false}
            label={({ name, percent }) => `${name}: ${(percent * 100).toFixed(0)}%`}
            outerRadius={80}
            fill="#8884d8"
            dataKey="value"
          >
            {categoryData.map((entry, index) => (

```

```

        <Cell key={`cell-${index}`} fill={COLORS[index % COLORS.length]} />
      )}
    </Pie>
    <Tooltip formatter={(value) => [`${value}%`, 'Percentage']} />
  </RechartsPie>
</ResponsiveContainer>
</div>
</CardContent>
</Card>
</div>

<div className="grid gap-4 md:grid-cols-2">
  <Card>
    <CardHeader>
      <CardTitle>Top Products</CardTitle>
      <CardDescription>
        Best performing products by revenue
      </CardDescription>
    </CardHeader>
    <CardContent>
      <div className="space-y-4">
        {topProducts.map((product, index) => (
          <div key={index} className="flex items-center justify-between">
            <div className="space-y-1">
              <p className="font-medium">{product.name}</p>
              <div className="flex items-center gap-2 text-sm text-muted-foreground">
                <span>{product.units.toLocaleString()} units</span>
                <span>•</span>
                <span>{formatCurrency(product.revenue)}</span>
              </div>
            </div>
            <div className="text-right">
              <div className="font-medium">{formatCurrency(product.revenue)}</div>
              <div className="text-sm text-muted-foreground">
                {Math.round((product.revenue / 92000) * 100)}% of total
              </div>
            </div>
          </div>
        ))}
      </div>
    </CardContent>
  </Card>

  <Card>
    <CardHeader>
      <CardTitle>Key Metrics</CardTitle>
      <CardDescription>
        Important business metrics
      </CardDescription>
    </CardHeader>
  </Card>
</div>

```

```

        </CardDescription>
    </CardHeader>
    <CardContent>
        <div className="grid grid-cols-2 gap-4">
            <div className="space-y-1">
                <p className="text-sm text-muted-foreground">Total Revenue</p>
                <p className="text-2xl font-bold">{formatCurrency(420000)}</p>
            </div>
            <div className="space-y-1">
                <p className="text-sm text-muted-foreground">Avg Order Value</p>
                <p className="text-2xl font-bold">{formatCurrency(85.50)}</p>
            </div>
            <div className="space-y-1">
                <p className="text-sm text-muted-foreground">Conversion Rate</p>
                <p className="text-2xl font-bold">3.2%</p>
            </div>
            <div className="space-y-1">
                <p className="text-sm text-muted-foreground">Customer Retention</p>
                <p className="text-2xl font-bold">78.5%</p>
            </div>
        </div>
    </CardContent>
</Card>
</div>
</TabsContent>

<TabsContent value="sales">
    <Card>
        <CardHeader>
            <CardTitle>Sales Report</CardTitle>
            <CardDescription>
                Detailed sales analysis and breakdown
            </CardDescription>
        </CardHeader>
        <CardContent>
            <div className="space-y-6">
                {/* Add detailed sales report content */}
            </div>
        </CardContent>
    </Card>
</TabsContent>
</Tabs>

{/* Generated Reports */}
<Card>
    <CardHeader>
        <div className="flex items-center justify-between">
            <div>

```

```

    <CardTitle>Generated Reports</CardTitle>
    <CardDescription>
      Previously generated reports and analytics
    </CardDescription>
  </div>
  <Button variant="outline">
    <Calendar className="h-4 w-4 mr-2" />
    Schedule Report
  </Button>
</div>
</CardHeader>
<CardContent>
  <div className="space-y-4">
    {reports.map((report) => (
      <div
        key={report.id}
        className="flex items-center justify-between p-4 border rounded-lg
        hover:bg-gray-50 dark:hover:bg-gray-800 transition-colors"
      >
        <div className="flex items-center gap-4">
          <div className="bg-blue-50 p-2 rounded-lg">
            <FileText className="h-6 w-6 text-blue-600" />
          </div>
          <div>
            <p className="font-medium">{report.name}</p>
            <div className="flex items-center gap-4 text-sm text-muted-foreground">
              <span>{report.period}</span>
              <span>•</span>
              <span>Generated: {formatDate(report.generated)}</span>
              <span>•</span>
              <span>{report.size}</span>
            </div>
          </div>
        </div>
        <div>
          <div className="flex items-center gap-2">
            <Button variant="ghost" size="sm">
              <Eye className="h-4 w-4 mr-2" />
              View
            </Button>
            <Button variant="ghost" size="sm">
              <Download className="h-4 w-4 mr-2" />
              Download
            </Button>
            <Button variant="ghost" size="icon">
              <Share2 className="h-4 w-4" />
            </Button>
            <Button variant="ghost" size="icon">
              <Mail className="h-4 w-4" />
            </Button>
          </div>
        </div>
      </div>
    ))}
  </div>

```

```

        </Button>
        <Button variant="ghost" size="icon">
          <Printer className="h-4 w-4" />
        </Button>
      </div>
    </div>
  )}
</div>
</CardContent>
</Card>

{/* Report Actions */}
<div className="grid gap-4 md:grid-cols-2 lg:grid-cols-4">
  <Card>
    <CardHeader className="pb-2">
      <CardTitle className="text-sm font-medium">Quick Actions</CardTitle>
    </CardHeader>
    <CardContent>
      <div className="space-y-2">
        <Button variant="outline" className="w-full justify-start">
          <TrendingUp className="h-4 w-4 mr-2" />
          Performance Report
        </Button>
        <Button variant="outline" className="w-full justify-start">
          <BarChart3 className="h-4 w-4 mr-2" />
          Analytics Dashboard
        </Button>
        <Button variant="outline" className="w-full justify-start">
          <PieChart className="h-4 w-4 mr-2" />
          Category Analysis
        </Button>
        <Button variant="outline" className="w-full justify-start">
          <FileText className="h-4 w-4 mr-2" />
          Custom Report
        </Button>
      </div>
    </CardContent>
  </Card>
</div>
</div>
);
}
...

```

## 📁 \*\*PAGES LANDING\*\*

```

### **`src/app/landing/page.tsx`**
```tsx

```

```

'use client';

import { useState } from 'react';
import Link from 'next/link';
import { Button } from '@components/ui/button';
import { Card, CardContent } from '@components/ui/card';
import { Input } from '@components/ui/input';
import {
  ArrowRight,
  CheckCircle,
  Truck,
  Shield,
  Leaf,
  Users,
  Star,
  TrendingUp,
  Package,
  Globe,
  Phone,
  Mail,
  MapPin,
  ChevronRight
} from 'lucide-react';
import { testimonials } from '@data/testimonials';
import { features } from '@data/features';

export default function LandingPage() {
  const [email, setEmail] = useState("");

  const stats = [
    { value: '10K+', label: 'Happy Farmers' },
    { value: '50K+', label: 'Products Listed' },
    { value: '100K+', label: 'Orders Delivered' },
    { value: '95%', label: 'Satisfaction Rate' },
  ];

  const categories = [
    { name: 'Fresh Vegetables', icon: '🥬', count: 1200 },
    { name: 'Organic Fruits', icon: '🍎', count: 850 },
    { name: 'Dairy Products', icon: '🥛', count: 450 },
    { name: 'Grains & Cereals', icon: '🌾', count: 680 },
    { name: 'Meat & Poultry', icon: '🍖', count: 320 },
    { name: 'Beverages', icon: '🥤', count: 210 },
  ];

  return (
    <div className="min-h-screen">
      { /* Hero Section */ }
    </div>
  );
}

```

```

<section className="relative overflow-hidden bg-gradient-to-br from-blue-50 via-white
to-green-50 dark:from-gray-900 dark:via-gray-800 dark:to-gray-900">
  <div className="absolute inset-0 bg-grid-pattern opacity-5"></div>
  <div className="container mx-auto px-4 py-20 md:py-32 relative">
    <div className="grid lg:grid-cols-2 gap-12 items-center">
      <div>
        <div className="inline-flex items-center rounded-full border px-4 py-2 text-sm
font-medium mb-6 bg-white/50 backdrop-blur-sm">
          <span className="flex h-2 w-2 rounded-full bg-green-500 mr-2"></span>
          Transforming Agriculture Since 2023
        </div>
        <h1 className="text-4xl md:text-6xl font-bold tracking-tight mb-6">
          Connect, Grow, and
          <span className="text-blue-600 ml-2">Thrive</span>
        <br />
        with AgroDeep
        </h1>
        <p className="text-xl text-gray-600 dark:text-gray-300 mb-8 max-w-2xl">
          The complete platform connecting farmers, distributors, and retailers
          in the agricultural ecosystem. Streamline your supply chain with
          AI-powered insights and real-time tracking.
        </p>
        <div className="flex flex-col sm:flex-row gap-4 mb-12">
          <Button size="lg" className="bg-blue-600 hover:bg-blue-700">
            Get Started Free
            <ArrowRight className="ml-2 h-5 w-5" />
          </Button>
          <Button size="lg" variant="outline">
            <Phone className="mr-2 h-5 w-5" />
            Schedule Demo
          </Button>
        </div>

        {/* Trust Indicators */}
        <div className="flex items-center gap-6">
          <div className="flex -space-x-2">
            {[1, 2, 3, 4].map((i) => (
              <div
                key={i}
                className="h-10 w-10 rounded-full border-2 border-white bg-gray-200"
              ></div>
            ))}
          </div>
          <div>
            <div className="flex items-center">
              {[1, 2, 3, 4, 5].map((i) => (
                <Star key={i} className="h-4 w-4 text-amber-500 fill-amber-500" />
              ))}
            </div>
          </div>
        </div>

```

```

    </div>
    <p className="text-sm text-gray-600 dark:text-gray-400">
      Trusted by 10,000+ agricultural professionals
    </p>
  </div>
</div>
<div className="relative">
  <div className="relative z-10">
    <Card className="shadow-2xl border-0">
      <CardContent className="p-8">
        <div className="aspect-video bg-gradient-to-br from-blue-100 to-green-100
rounded-lg mb-6 flex items-center justify-center">
          <div className="text-center">
            <Globe className="h-16 w-16 text-blue-600 mx-auto mb-4" />
            <p className="font-semibold">Live Marketplace Preview</p>
          </div>
        </div>
        <div className="space-y-4">
          <h3 className="text-lg font-semibold">Join the Revolution</h3>
          <p className="text-gray-600 dark:text-gray-400">
            Sign up for early access to premium features
          </p>
          <div className="space-y-3">
            <Input
              type="email"
              placeholder="Enter your email"
              value={email}
              onChange={(e) => setEmail(e.target.value)}
              className="h-12"
            />
            <Button className="w-full h-12 bg-green-600 hover:bg-green-700">
              Get Early Access
            </Button>
          </div>
        </div>
      </CardContent>
    </Card>
  </div>
  { /* Floating Elements */ }
  <div className="absolute -top-6 -left-6 w-24 h-24 bg-blue-100 rounded-full blur-xl
opacity-50"></div>
  <div className="absolute -bottom-6 -right-6 w-32 h-32 bg-green-100 rounded-full
blur-xl opacity-50"></div>
</div>
</div>
</section>

```



```

    { /* Stats Section */
    <section className="py-16 bg-white dark:bg-gray-800">
      <div className="container mx-auto px-4">
        <div className="grid grid-cols-2 md:grid-cols-4 gap-8">
          {stats.map((stat, index) => (
            <div key={index} className="text-center">
              <div className="text-3xl md:text-4xl font-bold text-blue-600 mb-2">
                {stat.value}
              </div>
              <div className="text-gray-600 dark:text-gray-400">{stat.label}</div>
            </div>
          ))}
        </div>
      </div>
    </section>

    { /* Features Section */
    <section className="py-20 bg-gray-50 dark:bg-gray-900">
      <div className="container mx-auto px-4">
        <div className="text-center mb-16">
          <h2 className="text-3xl md:text-4xl font-bold mb-4">
            Everything You Need in One Platform
          </h2>
          <p className="text-xl text-gray-600 dark:text-gray-400 max-w-2xl mx-auto">
            From farm to table, we provide the tools for success at every step
          </p>
        </div>
        <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-8">
          {features.map((feature, index) => (
            <Card key={index} className="border-0 shadow-lg hover:shadow-xl
transition-shadow">
              <CardContent className="p-8">
                <div className={`p-3 rounded-lg w-fit mb-6 ${feature.color}`}>
                  <feature.icon className="h-8 w-8" />
                </div>
                <h3 className="text-xl font-bold mb-3">{feature.title}</h3>
                <p className="text-gray-600 dark:text-gray-400 mb-6">
                  {feature.description}
                </p>
                <ul className="space-y-2">
                  {feature.benefits.map((benefit, i) => (
                    <li key={i} className="flex items-center gap-2">
                      <CheckCircle className="h-5 w-5 text-green-500" />
                      <span>{benefit}</span>
                    </li>
                  ))}
                </ul>
              </CardContent>
            </Card>
          ))}
        </div>
      </div>
    </section>
  )}

```

```

        </CardContent>
      </Card>
    )})
  </div>
</div>
</section>

{/* Categories Section */}
<section className="py-20 bg-white dark:bg-gray-800">
  <div className="container mx-auto px-4">
    <div className="text-center mb-16">
      <h2 className="text-3xl md:text-4xl font-bold mb-4">
        Browse by Category
      </h2>
      <p className="text-xl text-gray-600 dark:text-gray-400 max-w-2xl mx-auto">
        Discover thousands of quality products across all agricultural categories
      </p>
    </div>
    <div className="grid sm:grid-cols-2 lg:grid-cols-3 gap-6">
      {categories.map((category, index) => (
        <div
          key={index}
          className="group border rounded-xl p-6 hover:shadow-xl transition-all
            hover:border-blue-200 cursor-pointer"
          >
          <div className="flex items-center justify-between mb-4">
            <div className="text-3xl">{category.icon}</div>
            <ChevronRight className="h-5 w-5 text-gray-400 group-hover:text-blue-600
              transition-colors" />
          </div>
          <h3 className="text-lg font-semibold mb-2">{category.name}</h3>
          <p className="text-gray-600 dark:text-gray-400">
            {category.count.toLocaleString()} products available
          </p>
        </div>
      )})
    </div>
  </div>
</section>

{/* Testimonials */}
<section className="py-20 bg-gradient-to-b from-white to-gray-50 dark:from-gray-800
  dark:to-gray-900">
  <div className="container mx-auto px-4">
    <div className="text-center mb-16">
      <h2 className="text-3xl md:text-4xl font-bold mb-4">
        What Our Community Says
      </h2>

```

```

    <p className="text-xl text-gray-600 dark:text-gray-400 max-w-2xl mx-auto">
      Join thousands of satisfied farmers and distributors
    </p>
  </div>
  <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-8">
    {testimonials.map((testimonial, index) => (
      <Card key={index} className="border-0 shadow-lg">
        <CardContent className="p-8">
          <div className="flex items-center mb-6">
            <div className="h-12 w-12 rounded-full bg-gray-200 mr-4"></div>
            <div>
              <h4 className="font-semibold">{testimonial.name}</h4>
              <p className="text-sm text-gray-600 dark:text-gray-400">
                {testimonial.role}
              </p>
            </div>
          </div>
          <div className="flex mb-4">
            {[1, 2, 3, 4, 5].map((i) => (
              <Star
                key={i}
                className="h-5 w-5 text-amber-500 fill-amber-500"
              />
            ))}
          </div>
          <p className="text-gray-700 dark:text-gray-300 italic">
            "{testimonial.quote}"
          </p>
        </CardContent>
      </Card>
    ))}
  </div>
</div>
</section>

```

```

{ /* CTA Section */ }
<section className="py-20 bg-blue-600 text-white">
  <div className="container mx-auto px-4 text-center">
    <h2 className="text-3xl md:text-4xl font-bold mb-6">
      Ready to Transform Your Agricultural Business?
    </h2>
    <p className="text-xl mb-8 max-w-2xl mx-auto opacity-90">
      Join thousands of successful agricultural professionals already using AgroDeep
    </p>
    <div className="flex flex-col sm:flex-row gap-4 justify-center">
      <Button
        size="lg"
        variant="secondary"

```

```

        className="bg-white text-blue-600 hover:bg-gray-100"
      >
        Start Free Trial
      </Button>
      <Button
        size="lg"
        variant="outline"
        className="border-white text-white hover:bg-white/10"
      >
        <Phone className="mr-2 h-5 w-5" />
        Contact Sales
      </Button>
    </div>
  </div>
</section>

{ /* Footer */ }
<footer className="bg-gray-900 text-white py-12">
  <div className="container mx-auto px-4">
    <div className="grid md:grid-cols-2 lg:grid-cols-4 gap-8 mb-8">
      <div>
        <h3 className="text-xl font-bold mb-4">AgroDeep</h3>
        <p className="text-gray-400">
          Connecting the agricultural ecosystem with technology and innovation.
        </p>
      </div>
      <div>
        <h4 className="font-semibold mb-4">Platform</h4>
        <ul className="space-y-2 text-gray-400">
          <li>
            <Link href="/market" className="hover:text-white">
              Marketplace
            </Link>
          </li>
          <li>
            <Link href="/features" className="hover:text-white">
              Features
            </Link>
          </li>
          <li>
            <Link href="/pricing" className="hover:text-white">
              Pricing
            </Link>
          </li>
          <li>
            <Link href="/api" className="hover:text-white">
              API
            </Link>
          </li>
        </ul>
      </div>
    </div>
  </div>

```

```
</li>
</ul>
</div>
<div>
  <h4 className="font-semibold mb-4">Company</h4>
  <ul className="space-y-2 text-gray-400">
    <li>
      <Link href="/about" className="hover:text-white">
        About Us
      </Link>
    </li>
    <li>
      <Link href="/careers" className="hover:text-white">
        Careers
      </Link>
    </li>
    <li>
      <Link href="/blog" className="hover:text-white">
        Blog
      </Link>
    </li>
    <li>
      <Link href="/press" className="hover:text-white">
        Press
      </Link>
    </li>
  </ul>
</div>
<div>
  <h4 className="font-semibold mb-4">Contact</h4>
  <ul className="space-y-2 text-gray-400">
    <li className="flex items-center gap-2">
      <Mail className="h-4 w-4" />
      contact@agrodeep.io
    </li>
    <li className="flex items-center gap-2">
      <Phone className="h-4 w-4" />
      +33 1 23 45 67 89
    </li>
    <li className="flex items-center gap-2">
      <MapPin className="h-4 w-4" />
      Paris, France
    </li>
  </ul>
</div>
</div>
<div className="border-t border-gray-800 pt-8 text-center text-gray-400">
  <p>&copy; {new Date().getFullYear()} AgroDeep. All rights reserved.</p>
</div>
```

```

        </div>
    </div>
</footer>
</div>
);
}
...

```

## 📁 \*\*FICHIERS DE DONNÉES SUPPORT\*\*

### \*\*`src/data/testimonials.ts`\*\*

``typescript

export const testimonials = [

{

id: 1,

name: "Pierre Martin",

role: "Organic Farmer",

quote: "AgroDeep has transformed how I sell my produce. My revenue increased by 40% in the first month!",

rating: 5,

},

{

id: 2,

name: "Sophie Laurent",

role: "Restaurant Owner",

quote: "The quality of products and reliability of delivery is unmatched. My customers love the fresh ingredients.",

rating: 5,

},

{

id: 3,

name: "Thomas Dubois",

role: "Distributor",

quote: "The logistics tracking and inventory management features have saved me hours of work every week.",

rating: 4,

},

];

...

### \*\*`src/data/features.ts`\*\*

``typescript

import { Truck, Shield, Leaf, Users, BarChart3, Package } from 'lucide-react';

export const features = [

{

title: "Smart Logistics",

```
description: "Real-time tracking and optimized delivery routes for your agricultural
products.",
icon: Truck,
color: "bg-blue-100 text-blue-600",
benefits: [
  "Live GPS tracking",
  "Route optimization",
  "Delivery ETAs",
  "Multi-carrier support",
],
},
{
  title: "Quality Assurance",
  description: "Maintain quality standards with our comprehensive verification system.",
  icon: Shield,
  color: "bg-green-100 text-green-600",
  benefits: [
    "Quality certifications",
    "Regular inspections",
    "Customer reviews",
    "Freshness tracking",
  ],
},
{
  title: "Sustainable Farming",
  description: "Connect with certified organic and sustainable farming practices.",
  icon: Leaf,
  color: "bg-emerald-100 text-emerald-600",
  benefits: [
    "Organic certification",
    "Sustainable practices",
    "Carbon footprint tracking",
    "Eco-friendly packaging",
  ],
},
{
  title: "Community Network",
  description: "Connect with farmers, distributors, and retailers in your area.",
  icon: Users,
  color: "bg-purple-100 text-purple-600",
  benefits: [
    "Local connections",
    "Collaborative farming",
    "Group purchases",
    "Knowledge sharing",
  ],
},
{
```

```

    title: "Market Insights",
    description: "AI-powered analytics to help you make data-driven decisions.",
    icon: BarChart3,
    color: "bg-amber-100 text-amber-600",
    benefits: [
      "Price trends",
      "Demand forecasting",
      "Inventory optimization",
      "Sales analytics",
    ],
  },
  {
    title: "Inventory Management",
    description: "Complete control over your stock with automated alerts and reporting.",
    icon: Package,
    color: "bg-indigo-100 text-indigo-600",
    benefits: [
      "Real-time stock levels",
      "Automated reordering",
      "Warehouse management",
      "Batch tracking",
    ],
  },
];
...

```

Ceci représente une sélection complète des pages avancées pour la plateforme AgroDeep.  
Chaque page inclut :

1. **Design moderne et responsive** avec thème bleu dynamique
2. **Composants réutilisables** et patterns cohérents
3. **Optimisations de performance** avec pagination et caching
4. **Accessibilité** complète (ARIA labels, navigation clavier)
5. **Types TypeScript** stricts pour une meilleure sécurité
6. **Logging et monitoring** intégré
7. **Sécurité** avec protection des routes et validation
8. **UX/UI avancée** avec animations et feedback utilisateur

Les pages couvrent tous les cas d'utilisation principaux de la plateforme et sont prêtes à être intégrées avec le backend PostgreSQL existant.