

# TCF/TEF Platform - Development Guide

TCF/TEF Platform Team

2025-08-18

## Contents

<b>1</b>	<b>Development Guide - TCF/TEF Platform</b>	<b>2</b>
1.1	□ Overview . . . . .	2
1.2	□ Quick Start . . . . .	2
1.2.1	Prerequisites . . . . .	2
1.2.2	Installation . . . . .	2
1.3	□ Project Structure . . . . .	2
1.4	□ Development Workflow . . . . .	3
1.4.1	Code Style . . . . .	3
1.4.2	Git Workflow . . . . .	4
1.5	□ Testing . . . . .	4
1.5.1	Running Tests . . . . .	4
1.5.2	Writing Tests . . . . .	4
1.6	□ Database Management . . . . .	5
1.6.1	Prisma Setup . . . . .	5
1.7	□ Authentication . . . . .	6
1.7.1	NextAuth.js Setup . . . . .	6
1.8	□ Styling . . . . .	7
1.8.1	Tailwind CSS . . . . .	7
1.9	□ Scripts . . . . .	7
1.9.1	Available Commands . . . . .	7
1.10	□ Deployment . . . . .	7
1.10.1	Environment Variables . . . . .	7
1.10.2	Vercel Deployment . . . . .	8
1.10.3	Docker Deployment . . . . .	8
1.11	□ Debugging . . . . .	8
1.11.1	Common Issues . . . . .	8
1.11.2	VS Code Debugging . . . . .	9
1.12	□ Resources . . . . .	9
1.12.1	Documentation . . . . .	9
1.12.2	Tools . . . . .	9

# 1 Development Guide - TCF/TEF Platform

## 1.1 Overview

Complete development guide for setting up, developing, testing, and contributing to the TCF/TEF preparation platform.

## 1.2 Quick Start

### 1.2.1 Prerequisites

- **Node.js:** 18.0.0+
- **pnpm:** 8.0.0+ (or npm 9.0.0+)
- **PostgreSQL:** 15.0+
- **Git:** 2.30.0+

### 1.2.2 Installation

#### 1. Clone repository

```
git clone https://github.com/your-org/tcf-tef-platform.git
cd tcf-tef-platform
```

#### 2. Install dependencies

```
pnpm install
```

#### 3. Setup environment

```
cp .env.example .env.local
# Configure your environment variables
```

#### 4. Setup database

```
pnpm db:migrate
pnpm db:seed
```

#### 5. Start development

```
pnpm dev
```

#### 6. Open browser

Navigate to <http://localhost:3000>

## 1.3 Project Structure

```
tcf-tef-platform/
├── app/                # Next.js 13+ app directory
│   ├── admin/         # Admin section
│   ├── manager/       # Manager section
│   ├── api/           # API routes
│   ├── layout.tsx     # Root layout
│   └── page.tsx       # Home page
└── components/        # Reusable components
```

		ui/	# shadcn/ui components
		forms/	# Form components
	lib/		# Utility functions
	hooks/		# Custom React hooks
	types/		# TypeScript definitions
	prisma/		# Database schema
	docs/		# Documentation

## 1.4 📁 Development Workflow

### 1.4.1 Code Style

#### 1.4.1.1 TypeScript

```
interface ComponentProps {
  title: string;
  description?: string;
  onAction?: () => void;
}

export function Component({ title, description, onAction }: ComponentProps) {
  return (
    <div className="component">
      <h2>{title}</h2>
      {description && <p>{description}</p>}
      {onAction && <button onClick={onAction}>Action</button>}
    </div>
  );
}
```

#### 1.4.1.2 API Routes

```
import { NextApiRequest, NextApiResponse } from 'next';
import { withAuth } from '@lib/auth';

async function handler(req: NextApiRequest, res: NextApiResponse) {
  if (req.method !== 'GET') {
    return res.status(405).json({ error: 'Method not allowed' });
  }

  try {
    const data = await fetchData();
    return res.status(200).json(data);
  } catch (error) {
    return res.status(500).json({ error: 'Internal server error' });
  }
}
```

```
export default withAuth(handler);
```

## 1.4.2 Git Workflow

### 1.4.2.1 Branch Naming

```
feature/user-authentication  
bugfix/login-validation  
hotfix/security-patch
```

### 1.4.2.2 Commit Messages

```
feat(auth): add user authentication system  
fix(api): resolve login validation issue  
docs(readme): update installation instructions  
test(components): add unit tests for CourseCard
```

## 1.5 Testing

### 1.5.1 Running Tests

```
# All tests
```

```
pnpm test
```

```
# Watch mode
```

```
pnpm test:watch
```

```
# Coverage
```

```
pnpm test:coverage
```

```
# E2E tests
```

```
pnpm test:e2e
```

### 1.5.2 Writing Tests

#### 1.5.2.1 Component Testing

```
import { render, screen, fireEvent } from '@testing-library/react';  
import { CourseCard } from '../components/CourseCard';
```

```
describe('CourseCard', () => {  
  const mockCourse = {  
    id: '1',  
    title: 'French Grammar',  
    category: 'grammar'  
  };  
});
```

```
it('renders course information', () => {  
  render(<CourseCard course={mockCourse} />);
```

```

    expect(screen.getByText('French Grammar')).toBeInTheDocument();
  });

  it('handles enrollment click', () => {
    const mockEnroll = jest.fn();
    render(<CourseCard course={mockCourse} onEnroll={mockEnroll} />);

    fireEvent.click(screen.getByText('Enroll'));
    expect(mockEnroll).toHaveBeenCalledTimes(1);
  });
});

```

### 1.5.2.2 API Testing

```

import { createMocks } from 'node-mocks-http';
import handler from '../pages/api/courses';

describe('/api/courses', () => {
  it('returns courses list', async () => {
    const { req, res } = createMocks({ method: 'GET' });
    await handler(req, res);
    expect(res._getStatusCode()).toBe(200);
  });
});

```

## 1.6 Database Management

### 1.6.1 Prisma Setup

#### 1.6.1.1 Schema Example

```

// prisma/schema.prisma
model User {
  id          String    @id @default(cuid())
  email       String    @unique
  firstName   String
  lastName    String
  role        UserRole @default(STUDENT)
  createdAt   DateTime @default(now())
}

enum UserRole {
  STUDENT
  JUNIOR_MANAGER
  SENIOR_MANAGER
  ADMIN
}

```

### 1.6.1.2 Database Commands

*# Generate client*

```
pnpm prisma generate
```

*# Run migrations*

```
pnpm prisma migrate dev
```

*# Reset database*

```
pnpm prisma migrate reset
```

*# Open Prisma Studio*

```
pnpm prisma studio
```

## 1.7 Authentication

### 1.7.1 NextAuth.js Setup

```
// app/api/auth/[...nextauth]/route.ts
import NextAuth from 'next-auth';
import CredentialsProvider from 'next-auth/providers/credentials';
import { PrismaAdapter } from '@next-auth/prisma-adapter';
import { prisma } from '@lib/prisma';

export const authOptions = {
  adapter: PrismaAdapter(prisma),
  providers: [
    CredentialsProvider({
      name: 'credentials',
      credentials: {
        email: { label: 'Email', type: 'email' },
        password: { label: 'Password', type: 'password' }
      },
      async authorize(credentials) {
        // Your authentication logic
      }
    })
  ],
  callbacks: {
    async jwt({ token, user }) {
      if (user) token.role = user.role;
      return token;
    },
    async session({ session, token }) {
      if (token) session.user.role = token.role;
      return session;
    }
  }
}
```

```
};

const handler = NextAuth(authOptions);
export { handler as GET, handler as POST };
```

## 1.8 Styling

### 1.8.1 Tailwind CSS

```
// Component with Tailwind classes
export function Button({ children, variant = 'default' }: ButtonProps) {
  return (
    <button className={cn(
      "px-4 py-2 rounded-md font-medium transition-colors",
      variant === 'default' && "bg-blue-600 text-white hover:bg-blue-700",
      variant === 'secondary' && "bg-gray-200 text-gray-900 hover:bg-gray-300"
    )}>
      {children}
    </button>
  );
}
```

## 1.9 Scripts

### 1.9.1 Available Commands

```
{
  "scripts": {
    "dev": "next dev",
    "build": "next build",
    "start": "next start",
    "lint": "next lint",
    "test": "jest",
    "test:watch": "jest --watch",
    "test:e2e": "playwright test",
    "db:migrate": "prisma migrate dev",
    "db:seed": "prisma db seed",
    "db:studio": "prisma studio"
  }
}
```

## 1.10 Deployment

### 1.10.1 Environment Variables

```
DATABASE_URL="postgresql://user:password@localhost:5432/database"
NEXTAUTH_SECRET="your-secret"
NEXTAUTH_URL="http://localhost:3000"
```

```
AWS_ACCESS_KEY_ID="your-aws-key"
AWS_SECRET_ACCESS_KEY="your-aws-secret"
```

### 1.10.2 Vercel Deployment

1. Connect GitHub repository to Vercel
2. Configure environment variables
3. Deploy automatically on push to main

### 1.10.3 Docker Deployment

```
FROM node:18-alpine
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
RUN npm run build
EXPOSE 3000
CMD ["npm", "start"]
```

## 1.11 ☐ Debugging

### 1.11.1 Common Issues

#### 1.11.1.1 Database Connection

```
# Check PostgreSQL status
sudo systemctl status postgresql
```

```
# Reset Prisma
pnpm prisma generate
```

#### 1.11.1.2 Build Issues

```
# Clear cache
rm -rf .next
```

```
# Reinstall dependencies
rm -rf node_modules
pnpm install
```

#### 1.11.1.3 Authentication Issues

- Check environment variables
- Clear browser cookies
- Check network tab for errors



### 1.11.2 VS Code Debugging

```
// .vscode/launch.json
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Next.js: debug server-side",
      "type": "node-terminal",
      "request": "launch",
      "command": "npm run dev"
    },
    {
      "name": "Next.js: debug client-side",
      "type": "chrome",
      "request": "launch",
      "url": "http://localhost:3000"
    }
  ]
}
```

## 1.12 📁 Resources

### 1.12.1 Documentation

- [Next.js Docs](#)
- [React Docs](#)
- [TypeScript Docs](#)
- [Tailwind CSS Docs](#)
- [Prisma Docs](#)

### 1.12.2 Tools

- [VS Code](#)
- [Postman](#)
- [Prisma Studio](#)
- [Playwright](#)

---

**Document Version:** 1.0.0 **Last Updated:** January 2025