Installation Guide

Prerequisites

Before installing the Productivity Suite Web Application, ensure you have the following installed on your system:

Required Software

1. **Node.js** (version 18.0 or higher)
   - Download from nodejs.org
   - Verify installation: `node --version`
2. **npm** (usually comes with Node.js)
   - Verify installation: `npm --version`
   - Alternative: You can use **yarn** or **pnpm**
3. **Git** (for cloning the repository)
   - Download from git-scm.com
   - Verify installation: `git --version`

System Requirements

- **Operating System**: Windows 10+, macOS 10.15+, or Linux
- **RAM**: Minimum 4GB (8GB recommended)
- **Storage**: At least 500MB free space
- **Browser**: Chrome 90+, Firefox 88+, Safari 14+, or Edge 90+

Installation Steps

1. Clone the Repository

```
# Using HTTPS
git clone https://github.com/yourusername/notepad-web.git

# Or using SSH (if configured)
git clone git@github.com:yourusername/notepad-web.git

# Navigate to the project directory
cd notepad-web
```

2. Install Dependencies

```
# Using npm
npm install
```

```
# Or using yarn
yarn install

# Or using pnpm
pnpm install
```

3. Environment Setup

The application works out of the box without additional environment configuration. However, you can create a `.env.local` file for custom settings:

```
# Optional: Create environment file
touch .env.local
```

Example `.env.local` content:

```
# Development port (default: 3000)
VITE_PORT=3000

# Application title
VITE_APP_TITLE=″My Productivity Suite″
```

4. Start Development Server

```
# Start the development server
npm run dev

# Or with yarn
yarn dev

# Or with pnpm
pnpm dev
```

The application will be available at `http://localhost:3000`

Verification

Check Installation Success

1. **Server Running**: You should see output similar to:

   ```
   VITE v4.x.x ready in xxx ms

   ➜  Local:   http://localhost:3000/
   ➜  Network: use --host to expose
   ```

2. **Browser Access**: Open `http://localhost:3000` in your browser

3. **Features Test**: Verify all modules are working:

    - ✅ Notepad: Create and edit documents
    - ✅ AI Chat: Interface loads (functionality depends on AI service)
    - ✅ Pomodoro: Timer starts and stops
    - ✅ Projects: Create tasks and projects

Production Build

Build for Production

```
# Create production build
npm run build

# Or with yarn
yarn build

# Or with pnpm
pnpm build
```

Preview Production Build

```
# Preview the production build locally
npm run preview

# Or with yarn
yarn preview

# Or with pnpm
pnpm preview
```

Deploy Production Build

The `dist/` folder contains the production-ready files. You can deploy them to:

- **Static Hosting**: Netlify, Vercel, GitHub Pages
- **CDN**: AWS CloudFront, Cloudflare
- **Web Server**: Apache, Nginx

Troubleshooting

Common Issues

1. Port Already in Use

Error: `Port 3000 is already in use`

Solution:

```
# Kill process using port 3000
npx kill-port 3000

# Or use a different port
npm run dev -- --port 3001
```

2. Node Version Issues

Error: `Node version not supported`

Solution:

```
# Check current Node version
node --version

# Update Node.js to latest LTS version
# Visit nodejs.org for installer

# Or use Node Version Manager (nvm)
nvm install --lts
nvm use --lts
```

3. Dependency Installation Fails

Error: `npm install` fails

Solution:

```
# Clear npm cache
npm cache clean --force

# Delete node_modules and package-lock.json
rm -rf node_modules package-lock.json

# Reinstall dependencies
npm install
```

4. Build Errors

Error: TypeScript or build errors

Solution:

```
# Check for TypeScript errors
npm run type-check
```

```
# Run linter
npm run lint

# Fix auto-fixable issues
npm run lint -- --fix
```

5. Browser Compatibility

Issue: Application not working in older browsers

Solution:

- Update to a supported browser version
- Enable JavaScript if disabled
- Clear browser cache and cookies

Performance Issues

Slow Development Server

```
# Increase Node.js memory limit
node --max-old-space-size=4096 node_modules/.bin/vite

# Or set in package.json scripts
"dev": "node --max-old-space-size=4096 node_modules/.bin/vite"
```

Large Bundle Size

```
# Analyze bundle size
npm run build -- --analyze

# Check for unused dependencies
npx depcheck
```

Development Setup

IDE Configuration

VS Code (Recommended)

Install these extensions:

- ES7+ React/Redux/React-Native snippets
- TypeScript Importer
- Tailwind CSS IntelliSense
- Prettier - Code formatter
- ESLint

VS Code Settings

Create `.vscode/settings.json`:

```
{
  "editor.formatOnSave": true,
  "editor.defaultFormatter": "esbenp.prettier-vscode",
  "typescript.preferences.importModuleSpecifier": "relative",
  "tailwindCSS.experimental.classRegex": [
    ["cva\\(([^)]*)\\)", "[\"'`]([^\"'`]*).*?[\"'`]"],
    ["cx\\(([^)]*)\\)", "(?:'|\"|`)([^']*)(?:'|\"|`)"],
    ["cn\\(([^)]*)\\)", "(?:'|\"|`)([^']*)(?:'|\"|`)"]
  ]
}
```

Git Hooks (Optional)

Set up pre-commit hooks:

```
# Install husky
npm install --save-dev husky

# Initialize husky
npx husky install

# Add pre-commit hook
npx husky add .husky/pre-commit "npm run lint && npm run type-check"
```

Database Setup

IndexedDB

The application uses IndexedDB for local storage. No additional setup required:

- **Automatic**: Database created on first use
- **Browser Storage**: Data stored locally in browser
- **Capacity**: Typically 50MB+ depending on browser

Data Migration

If upgrading from an older version:

1. **Backup**: Export your data before upgrading
2. **Clear Storage**: Clear browser data if issues occur
3. **Import**: Re-import your backed-up data

Security Setup

Content Security Policy (CSP)

For production deployment, add CSP headers:

```
<meta http-equiv=”Content-Security-Policy”
      content=”default-src ’self’;
               script-src ’self’ ’unsafe-inline’;
               style-src ’self’ ’unsafe-inline’;
               img-src ’self’ data: https:;”>
```

HTTPS Configuration

For production:

- Always use HTTPS
- Configure SSL certificates
- Enable HSTS headers

Monitoring Setup

Error Tracking

Integrate error tracking (optional):

```
# Install Sentry (example)
npm install @sentry/react @sentry/tracing
```

Analytics

Add analytics (optional):

```
# Install Google Analytics (example)
npm install gtag
```

Backup and Recovery

Data Backup

1. **Export Feature**: Use built-in export functionality
2. **Browser Backup**: Export IndexedDB data
3. **Regular Backups**: Set up automated exports

Recovery Process

1. **Fresh Install**: Follow installation steps
2. **Import Data**: Use import functionality
3. **Verify**: Check all data imported correctly

Support

Getting Help

- **Documentation**: Check README.md and TECHNICAL_DOCUMENTATION.md
- **Issues**: Create GitHub issues for bugs

- **Discussions**: Use GitHub Discussions for questions
- **Community**: Join our Discord/Slack community

Reporting Issues

When reporting issues, include:

- Operating system and version
- Node.js version
- Browser and version
- Error messages and stack traces
- Steps to reproduce

Next Steps: After successful installation, check out the README.md for usage instructions and the TECHNICAL_DOCUMENTATION.md for development details.