Installation:

To create a new project using MERN-X, simply run the following
command:

```
npx mern-x@latest
// For yarn
yarn exec mern-x
```

How to run project

```
npx run mern    // Project Back End Will Run @3000
npx run dev     // Complete Project Will Run @3001

// For yarn
yarn run mern
yarn run dev
```

How create Model

```
npm run create:model YourModelName
// For yarn
yarn run create:model YourModelName
```

How create Controller

```
npm run create:controller YourControllerName
// For yarn
yarn run create:controller YourControllerName
```

How create Middleware

```
npm run create:middleware YourMiddlewareName
// For yarn
yarn run create:middleware YourMiddlewareName
```

How create Page

```
npm run create:page YourPageName
// For yarn
yarn run create:page YourPageName
```

How create Component

```
npm run create:component YourComponentName
// For yarn
yarn run create:component YourComponentName
```

How create Loader

```
npm run create:loader YourLoaderName
// For yarn
yarn run create:loader YourLoaderName
```

How create Layout

```
npm run create:layout YourLayoutName
// For yarn
yarn run create:layout YourLayoutName
```

Project Structure

```
mern-x/
│
├── app/
│   ├── config/
│   │   ├── cli.js
│   │   ├── config.js
│   │
│   ├── controllers
│   │   ├── todoController.js
│   │
│   ├── middlewares
│   │   ├── authMiddleware.js
│   │
```

```
|       ├── models
|       |   ├── todosModel.js
|       |   |
|       ├── storage/
|       |   |
|       ├── utility/
|       |   ├── emailUtility.js
|       |   ├── tokenUtility.js
|       |   ├── validationUtility.js
|       |   |
|       ├── dist/
|       |   |
|       ├── node_modules/
|       |   |
|       ├── public/
|       |   |
|       ├── routes/
|       |   ├── api.js
|       |   ├── web.jsx
|       |   |
|       ├── views/
|       |   ├── assets/
|       |   |   ├── css/
|       |   |   |   ├── style.js
|       |   |
|       |   ├── components/
|       |   |   ├── CreateForm.jsx
|       |   |   ├── List.jsx
|       |   |
|       |   ├── layout/
|       |   |   ├── AppLayout.jsx
|       |   |   |
|       |   |
|       |   ├── loader/
|       |   |   ├── ListLoader.jsx
|       |   |   |
|       |   ├── pages/
|       |   |   ├── CreatePage.jsx
|       |   |   ├── ListPage.jsx
|       |   |
|       |   ├── main.jsx
|       |   |
|       |   |
|       ├── .gitattributes
|       ├── .gitignore
|       ├── app.js
```

```
|       ├── index.html
|       ├── LICENSE
|       ├── package.json
|       ├── package-lock.json
|       ├── postcss.config.js
|       ├── README.md
|       ├── tailwind.config.js
|       ├── vite.config.js
```

- app: This directory contains the main application code.
  - config: Configuration files for the application.
  - controllers: Controllers for handling business logic.
  - Middleware functions for handling requests.
  - models: Database models.
  - storage: Directory for storing files (if applicable).
  - utility: Utility functions.
- dist: Distribution files (compiled code).
- node_modules: Node.js modules installed via npm.
- public: Public assets.
- routes: Route definitions.
- views: React components and views.
  - assets: Static assets like CSS files.
  - components: Reusable React components.
  - layout: Layout components.
  - loader: Loading components.
  - pages: Individual pages/components.
- .gitattributes: Git attributes file.
- .gitignore: Git ignore file.
- app.js: Entry point of the Node.js application.
- index.html: Main HTML file.
- LICENSE: License file.
- package.json: Project metadata and dependencies.
- package-lock.json: Dependency lock file.
- postcss.config.js: PostCSS configuration file.
- README.md: Project documentation. -tailwind.config.js: Tailwind CSS configuration file. -vite.config.js: Vite configuration file.