



Kosmora

User Manual

MongoDB Schema Designer for VS Code

Version 1.0.0

© 2026 Sanket Santosh Vidya Manohar. All rights reserved.

sanket-manohar.github.io/kosmora

Contents

1. Introduction
2. Installing Kosmora
3. Quick start
4. The schema designer
5. Exporting a schema
6. Using your exports
7. Settings
8. Troubleshooting
9. License and support

Kosmora
v1.0.0 · crafted by Sanket Santosh Vidya Manohar

1 Introduction

Kosmora is an extension for Visual Studio Code that lets you draw your MongoDB data model on a canvas and export it as JSON Schema, MongoDB validators, Mongoose, TypeScript, Prisma, Pydantic, Markdown, Mermaid, DBML, SVG or PNG.

Completely offline

Kosmora never contacts the internet and never connects to a database. It has no telemetry and loads no online resources. It only reads and writes the design files you open and the files you export.

Words used in this guide

Term	Meaning
Schema design	A <code>.kosmora.json</code> file that describes your collections and fields.
Collection	A group of documents in MongoDB, like a table in SQL.
Field	A named value inside a document, such as <code>email</code> or <code>price</code> .
Reference	A field that stores the <code>_id</code> of a document in another collection.
Command Palette	The VS Code command box. Open it with <code>Ctrl</code> + <code>Shift</code> + <code>P</code> (<code>Cmd</code> + <code>Shift</code> + <code>P</code> on macOS).

2 Installing Kosmora

You need Visual Studio Code 1.85 or newer on Windows, macOS or Linux.

From the Marketplace

1. Open VS Code and click the **Extensions** icon in the activity bar, or press `Ctrl + Shift + X`.
2. Search for **Kosmora**.
3. Click **Install** on "Kosmora: MongoDB Schema Designer" by sanket-manohar.

From a VSIX file (offline install)

1. Download `kosmora-1.0.0.vsix` from the [GitHub releases page](#).
2. In VS Code, open the Extensions view, click the `...` menu at its top and choose **Install from VSIX....**
3. Pick the downloaded file. Kosmora is ready when the planet icon appears in the activity bar.

You can also install from a terminal:

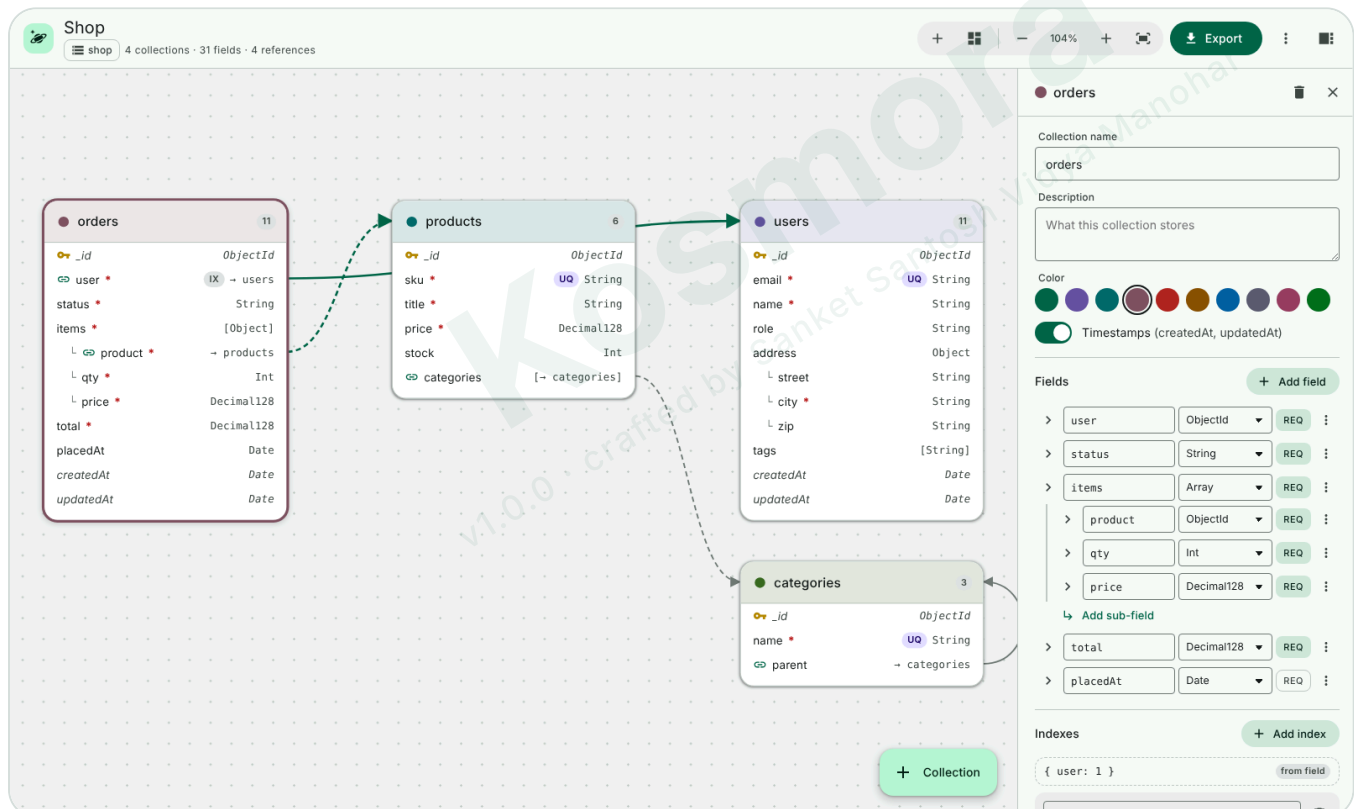
```
code --install-extension kosmora-1.0.0.vsix
```

3 Quick start

Kosmora includes an interactive walkthrough with screenshots. It opens on VS Code's Welcome page after you install, and you can reopen it at any time with **Kosmora: Get Started with Kosmora** from the Command Palette or the **⋮** menu of the Kosmora sidebar.

This five-minute tour creates a design and exports it.

1. Open a folder in VS Code (**File → Open Folder...**).
2. Open the Command Palette and run **Kosmora: New Sample Schema (Shop)**. A ready-made shop design opens in the designer.
3. Click a collection card, such as **users**. The panel on the right shows its fields. Change a field name or type and watch the card update.
4. Click **Export** in the top bar and choose a format, for example **Mongoose models**. The generated code opens in a new editor.
5. Click the planet icon in the activity bar. The **Schema Designs** view lists every design in your workspace.



The sample Shop schema in the designer.

4 The schema designer

Creating and opening designs

- Run **Kosmora: New Schema Design** to start an empty design, or **Kosmora: New Sample Schema (Shop)** to start from an example.
- Any file whose name ends in `.kosmora.json` opens in the designer automatically. The **Schema Designs** view in the Kosmora sidebar lists them all; click one to open it.
- To see the raw JSON, choose **Open as JSON** from the **More actions** menu (⋮). To switch back, run **Kosmora: Open in Schema Designer**.
- Designs save like any file with `Ctrl + S`, and undo and redo work as usual.

The screen at a glance

Area	What it does
Top bar	Design name, database name and counts; add collection, auto-arrange, zoom, fit, Export , the More actions menu and the inspector toggle.
Canvas	Your collections as cards. Arrows show references; dashed arrows mean one-to-many (an array of references).
Inspector	The right-hand panel. It shows schema details when nothing is selected, or the selected collection's settings, fields and indexes.

Collections

1. Click the **+ Collection** button, press `A`, or double-click an empty part of the canvas.
2. Type a name in **Collection name**. Add a **Description** if you like; it appears in docs and generated code.
3. Pick a **Color** to group related collections.
4. Turn on **Timestamps** to add `createdAt` and `updatedAt` automatically.
5. Drag a card by its header to move it. Press `L` to auto-arrange everything.

Fields

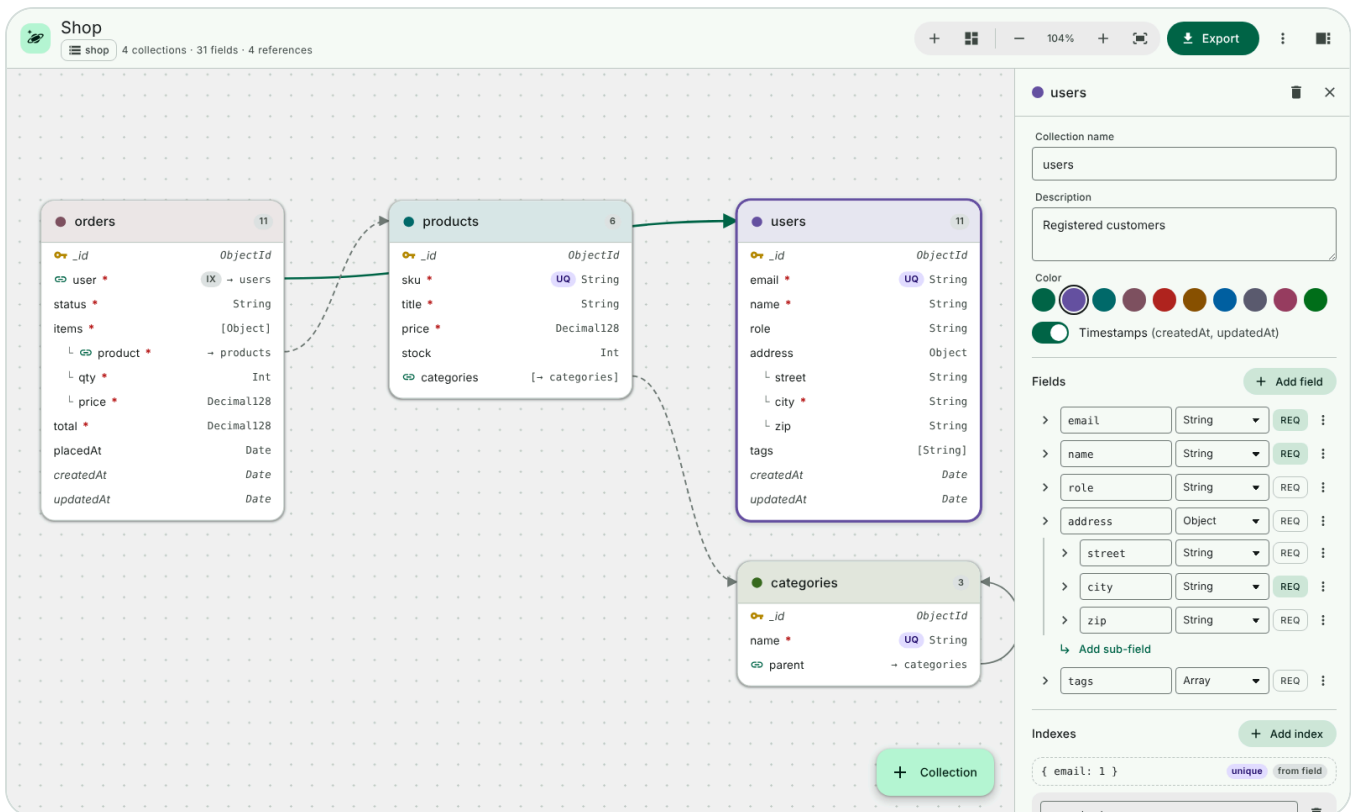
1. Select a collection and click **Add field**.
2. Type the field name and choose its type.
3. Click **REQ** to make the field required.
4. Click the arrow (**Show details**) for more options: **Unique**, **Indexed**, **Default**, **Description**, **Minimum/Maximum**, **Min length/Max length**, **Pattern (regex)** and **Allowed values**.
5. Use the  menu on a field to move it up or down, duplicate it or delete it.

`_id` is added automatically as an ObjectId, so you don't need to create it.

Type	Use it for
String	Text
Number, Int, Long, Double, Decimal128	Numbers. Use Decimal128 for money.
Boolean	true or false
Date	Dates and times
ObjectId	IDs, and references to other collections
UUID, Binary	UUIDs and binary data
Object	An embedded sub-document. Use Add sub-field to fill it.
Array	A list. Choose what it holds in Array of .
Mixed	Any value

References between collections

1. Add a field with type **ObjectId**, for example `user` in `orders`.
2. Open its details and choose the target in **References collection**, for example `users`.
3. An arrow appears between the cards. For one-to-many, make the field an **Array** of ObjectId with a reference; the arrow is dashed.



Editing a collection in the inspector.

Indexes

Marking a field **Unique** or **Indexed** creates a single-field index. For compound, text or TTL indexes, click **Add index** in the inspector, list the fields and choose the options.

Problems

Kosmora checks your design as you work. Duplicate names, invalid field names and references to collections that don't exist show a warning icon on the card and a count at the bottom left of the canvas. Click the count to see the list, and click a problem to select its collection.

Keyboard shortcuts

Key	Action
A	Add a collection
L	Auto-arrange
F	Fit everything on screen
+ / -	Zoom in or out (or Ctrl + scroll)
Del	Delete the selected collection
Esc	Clear the selection

Kosmora
v1.0.0 · crafted by Sanket Santosh Vidya Manohar

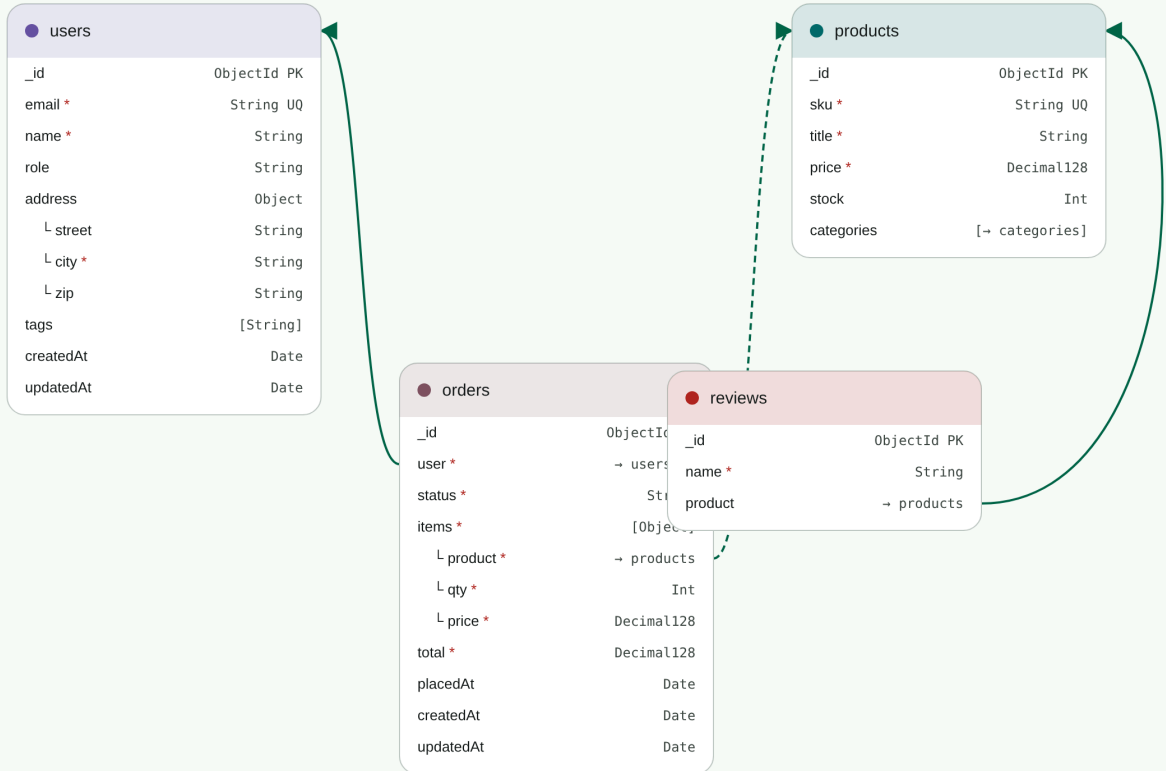
5 Exporting a schema

Click **Export** in the designer and choose a format. Or use the Command Palette:

- **Kosmora: Export Schema...** saves one format to a file.
- **Kosmora: Export Schema in All Formats...** saves every format into a folder you choose.
- **Kosmora: Preview Schema Export...** opens the result in an unsaved editor so you can look before saving.

Format	What you get
JSON Schema	Draft 2020-12 definitions, one per collection.
MongoDB validators	A mongosh script that creates each collection with a <code>\$jsonSchema</code> validator, plus its indexes.
Mongoose models	Schemas with refs, validation and indexes, as CommonJS or ES modules.
TypeScript interfaces	Typed documents using the <code>mongoose</code> package's types.
Prisma schema	The MongoDB provider, with composite types and relations.
Pydantic models	Python Pydantic v2 models.
Markdown docs	Field tables, indexes and a Mermaid diagram.
Mermaid	An <code>erDiagram</code> you can paste into GitHub or docs.
DBML	For dbdiagram.io and similar tools.
SVG and PNG	Images of your diagram, in light or dark.

Shop



A diagram exported as PNG.

6 Using your exports

Kosmora never connects to a database. It only writes the files you export, and you decide where to use them.

- **Application code.** Copy the Mongoose models, TypeScript interfaces, Prisma schema or Pydantic models into your project and adjust them as needed.
- **Documentation.** Commit the Markdown, Mermaid, DBML or image exports next to your code, or paste them into a wiki.
- **MongoDB validators.** The exported `.mongodb.js` file is a mongosh script that creates each collection with its validator and indexes. Read it first, then run it yourself on a test database:

```
mongosh "<your connection string>" shop.mongodb.js
```

Review before you run

Generated code and scripts are a starting point. Check them, and try them on a non-production copy of your data, before you use them on a real database. Keep backups of your data.

Kosmora
v1.0.0 · crafted by Sanket Santosh Vidya Manohar

7 Settings

Open **File** → **Preferences** → **Settings** and search for **Kosmora**.

Setting	Default	What it does
<code>kosmora.theme</code>	auto	Color scheme: auto (follow VS Code), light or dark.
<code>kosmora.export.outputFolder</code>	(empty)	Export folder, relative to the design file.
<code>kosmora.export.mongooseModuleSystem</code>	commonjs	CommonJS or ES module output for Mongoose.
<code>kosmora.export.imageTheme</code>	light	Theme of exported SVG and PNG diagrams.

Kosmora
v1.0.0 · crafted by Sanket Santosh Vidya Manohar

8 Troubleshooting

The designer opens as plain text

The file name must end in `.kosmora.json`. Right-click the file and choose **Open With... → Kosmora Schema Designer**, or run **Kosmora: Open in Schema Designer**.

I can't find my exported files

Kosmora asks where to save every export. The save dialog starts in the folder of the design file, or in the folder set in `kosmora.export.outputFolder`.

The diagram image is too dark or too light

Change `kosmora.export.imageTheme` to light or dark and export again.

Kosmora
v1.0.0 · crafted by Sanket Santosh Vidya Manohar

9 License and support

Kosmora is proprietary software. © 2026 Sanket Santosh Vidya Manohar. All rights reserved.

- **You may** download Kosmora for free from the official Marketplace listing or GitHub releases, and use it for personal, educational and commercial work. Everything you create with it is yours.
- **Your data:** Kosmora never connects to a database. You are responsible for how you use the code and scripts it generates, and for your backups. Kosmora is provided "as is", and its owner accepts no liability for data loss, corruption, downtime or other damage from using it or its output.
- **You may not** republish, redistribute, modify, update, sell or reverse engineer Kosmora, or use its name or logo for another product, without written permission.

Read the full terms in [LICENSE.md](#).

For help, bug reports and feature ideas, open an issue at github.com/sanket-manohar/kosmora/issues.

Trademarks and independence

Kosmora is a separate, independently developed tool. It is not affiliated with, sponsored by, endorsed by or connected to MongoDB, Inc., Microsoft Corporation or any other company whose products it designs for or exports to.

MongoDB, MongoDB Atlas and MongoDB Compass are trademarks of MongoDB, Inc. Visual Studio Code is a trademark of Microsoft Corporation. Mongoose, Prisma, Pydantic, TypeScript, Mermaid, DBML and all other names in this guide belong to their respective owners. They are mentioned only to describe what Kosmora designs for or exports to. Kosmora does not include, sell or connect to any of those products; you obtain and use them separately under their own terms.