

```
npm init -y
npm install express mongoose dotenv jsonwebtoken bcryptjs joi cors helmet morgan
npm install --save-dev typescript ts-node nodemon @types/node @types/express @types/jsonwebtoken
@types/mongoose @types/bcryptjs @types/joi
```

```
npm run dev
```

```
.env file
```

```
PORT=3000
```

mongoDB connection string

```
MONGO_URI=mongodb+srv://yashvendrajnv95:q7q9CVAsJY2je8KF*****
```

JWT secret keys

```
JWT_SECRET=123456789abc
```

```
JWT_REFRESH_SECRET=123456789abc
```

```
import mongoose, { Document, Types } from "mongoose";
```

```
export interface IBook extends Document {
```

```
  title: string;
```

```
  author: string;
```

```
  genre: string;
```

```
  description?: string;
```

```
  averageRating: number;
```

```
  createdBy: Types.ObjectId;
```

```
}
```

```
const bookSchema = new mongoose.Schema(  
{
```

```
  title: { type: String, required: true },
```

```
  author: { type: String, required: true },
```

```
  genre: { type: String, required: true },
```

```
  description: { type: String },
```

```
  averageRating: { type: Number, default: 0 },
```

```
  createdBy: { type: mongoose.Schema.Types.ObjectId, ref: "User", required: true },
```

```
},
```

```
{ timestamps: true }
```

```
);
```

```
export default mongoose.model("Book", bookSchema);
```

```
import mongoose, { Document, Types } from "mongoose";
```

```

export interface IReview extends Document {
  book: Types.ObjectId;
  user: Types.ObjectId;
  rating: number;
  comment: string;
}

const reviewSchema = new mongoose.Schema(
{
  book: { type: mongoose.Schema.Types.ObjectId, ref: "Book", required: true },
  user: { type: mongoose.Schema.Types.ObjectId, ref: "User", required: true },
  rating: { type: Number, min: 1, max: 5, required: true },
  comment: { type: String, required: true },
},
{ timestamps: true }
);

// One review per user per book
reviewSchema.index({ book: 1, user: 1 }, { unique: true });

export default mongoose.model("Review", reviewSchema);

import mongoose, { Document } from "mongoose";

export interface IUser extends Document {
  firstName: string;
  lastName: string;
  email: string;
  password: string;
}

const userSchema = new mongoose.Schema(
{
  firstName: { type: String, required: true, unique: true },
  lastName: { type: String, required: true, unique: true },
  email: { type: String, required: true, unique: true },
  password: { type: String, required: true },
},
{ timestamps: true }
);

export default mongoose.model("User", userSchema);

+-----+ +-----+ +-----+
| User | <-----> | Review | <-----> | Book |
+-----+ +-----+ +-----+
|| ^ |

```

| creates | | references | createdBy

+-----+ +-----+

Entity	Related To	Type	Description
User	Book	1-to-Many	One user can create many books
User	Review	1-to-Many	One user can post many reviews
Book	Review	1-to-Many	One book can have many reviews
Review	Book+User	Many-to-1	Each review belongs to 1 book and 1 user

```
{
  "info": {
    "_postman_id": "c742ff05-0dd0-4d99-bdf6-ef30795811a1",
    "name": "Billeasy",
    "schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json",
    "_exporter_id": "7707124"
  },
  "item": [
    {
      "name": "http://localhost:3000/api/v1/books",
      "request": {
        "method": "POST",
        "header": [
          {
            "key": "Authorization",
            "value": "Bearer $TOKEN"
          },
          {
            "key": "Content-Type",
            "value": "application/json"
          }
        ],
        "body": {
          "mode": "raw",
          "raw": "{\r\n  \"title\": \"The Alchemist\",\r\n  \"author\": \"Paulo Coelho\",\r\n  \"genre\": \"Fiction\",\r\n  \"publishedYear\": 1988\r\n}",
          "options": {
            "raw": {
              "language": "json"
            }
          }
        }
      },
      "url": {
        "raw": "http://localhost:3000/api/v1/books",
        "protocol": "http",
```

```

"host": [
  "localhost"
],
"port": "3000",
"path": [
  "api",
  "v1",
  "books"
],
},
"description": "Generated from cURL: curl -X POST http://localhost:3000/api/v1/books \\r\\n-H
\"Authorization: Bearer $TOKEN\" \\r\\n-H \"Content-Type: application/json\" \\r\\n-d '{\\r\\n \"title\": \"The
Alchemist\",\\r\\n \"author\": \"Paulo Coelho\",\\r\\n \"genre\": \"Fiction\",\\r\\n \"publishedYear\": 1988\\r\\n}'\\r\\n"
},
"response": []
},
{
"name": "http://localhost:3000/api/v1/books?
page=1&limit=10&author=Paulo%20Coelho&genre=Fiction",
"request": {
"method": "GET",
"header": [],
"url": {
"raw": "http://localhost:3000/api/v1/books?page=1&limit=10&author=Paulo%20Coelho&genre=Fiction",
"protocol": "http",
"host": [
"localhost"
],
"port": "3000",
"path": [
"api",
"v1",
"books"
],
"query": [
{
"key": "page",
"value": "1"
},
{
"key": "limit",
"value": "10"
},
{
"key": "author",

```

```

"value": "Paulo%20Coelho"
},
{
  "key": "genre",
  "value": "Fiction"
}
],
},
"description": "Generated from cURL: curl \"http://localhost:3000/api/v1/books?
page=1&limit=10&author=Paulo%20Coelho&genre=Fiction\"\\r\\n"
},
"response": []
},
{
  "name": "http://localhost:3000/api/v1/books/BOOK_ID_HERE",
  "request": {
    "method": "GET",
    "header": [],
    "url": {
      "raw": "http://localhost:3000/api/v1/books/BOOK_ID_HERE",
      "protocol": "http",
      "host": [
        "localhost"
      ],
      "port": "3000",
      "path": [
        "api",
        "v1",
        "books",
        "BOOK_ID_HERE"
      ]
    },
    "description": "Generated from cURL: curl http://localhost:3000/api/v1/books/BOOK_ID_HERE\\r\\n"
  },
  "response": []
},
{
  "name": "http://localhost:3000/api/v1/books/search/query?q=alchemist",
  "request": {
    "method": "GET",
    "header": [],
    "url": {
      "raw": "http://localhost:3000/api/v1/books/search/query?q=alchemist",
      "protocol": "http",
      "host": [

```

```

"localhost"
],
"port": "3000",
"path": [
"api",
"v1",
"books",
"search",
"query"
],
"query": [
{
"key": "q",
"value": "alchemist"
}
],
},
"description": "Generated from cURL: curl "http://localhost:3000/api/v1/books/search/query?
q=alchemist"\r\n"
},
"response": []
},
{
"name": "http://localhost:3000/api/v1/books/BOOK_ID_HERE/reviews",
"request": {
"method": "POST",
"header": [
{
"key": "Authorization",
"value": "Bearer $TOKEN"
},
{
"key": "Content-Type",
"value": "application/json"
}
],
"body": {
"mode": "raw",
"raw": "{\r\n  \"rating\": 5,\r\n  \"comment\": \"An inspiring read!\"\r\n}",
"options": {
"raw": {
"language": "json"
}
}
},
},
},

```

```

"url": {
"raw": "http://localhost:3000/api/v1/books/BOOK_ID_HERE/reviews",
"protocol": "http",
"host": [
"localhost"
],
"port": "3000",
"path": [
"api",
"v1",
"books",
"BOOK_ID_HERE",
"reviews"
]
},
"description": "Generated from cURL: curl -X POST
http://localhost:3000/api/v1/books/BOOK_ID_HERE/reviews \\r\\n-H \"Authorization: Bearer $TOKEN\" \\r\\n-
H \"Content-Type: application/json\" \\r\\n-d '{\\r\\n \"rating\": 5,\\r\\n \"comment\": \"An inspiring read!\"\\r\\n}'\\r\\n"
},
"response": []
},
{
"name": "http://localhost:3000/api/v1/reviews/REVIEW_ID_HERE",
"request": {
"method": "PUT",
"header": [
{
"key": "Authorization",
"value": "Bearer $TOKEN"
},
{
"key": "Content-Type",
"value": "application/json"
}
],
"body": {
"mode": "raw",
"raw": "{\\r\\n \"rating\": 4,\\r\\n \"comment\": \"Actually, I liked it slightly less on the second read.\"\\r\\n}",
"options": {
"raw": {
"language": "json"
}
}
},
"url": {

```

```

"raw": "http://localhost:3000/api/v1/reviews/REVIEW_ID_HERE",
"protocol": "http",
"host": [
  "localhost"
],
"port": "3000",
"path": [
  "api",
  "v1",
  "reviews",
  "REVIEW_ID_HERE"
],
"description": "Generated from cURL: curl -X PUT http://localhost:3000/api/v1/reviews/REVIEW_ID_HERE \\r\\n-H \"Authorization: Bearer $TOKEN\" \\r\\n-H \"Content-Type: application/json\" \\r\\n-d '{\\r\\n \"rating\": 4,\\r\\n \"comment\": \"Actually, I liked it slightly less on the second read.\"\\r\\n}'\\r\\n\"",
"response": []
},
{
  "name": "http://localhost:3000/api/v1/reviews/REVIEW_ID_HERE",
  "request": {
    "method": "DELETE",
    "header": [
      {
        "key": "Authorization",
        "value": "Bearer $TOKEN"
      }
    ],
    "url": {
      "raw": "http://localhost:3000/api/v1/reviews/REVIEW_ID_HERE",
      "protocol": "http",
      "host": [
        "localhost"
      ],
      "port": "3000",
      "path": [
        "api",
        "v1",
        "reviews",
        "REVIEW_ID_HERE"
      ]
    },
    "description": "Generated from cURL: curl -X DELETE http://localhost:3000/api/v1/reviews/REVIEW_ID_HERE \\r\\n-H \"Authorization: Bearer $TOKEN\"\\r\\n"
  }
}

```



```

},
"response": []
},
{
"name": "http://localhost:3000/api/v1/users/register",
"request": {
"method": "POST",
"header": [
{
"key": "Content-Type",
"value": "application/json"
}
],
"body": {
"mode": "raw",
"raw": "{\r\n \"firstName\": \"Yash\", \r\n \"lastName\": \"Kumar\", \r\n \"email\": \"yash12345999@gmail.com\", \r\n \"password\": \"Yash123@#\" \r\n}",
"options": {
"raw": {
"language": "json"
}
}
},
"url": {
"raw": "http://localhost:3000/api/v1/users/register",
"protocol": "http",
"host": [
"localhost"
],
"port": "3000",
"path": [
"api",
"v1",
"users",
"register"
]
},
"description": "Generated from cURL: curl --location 'http://localhost:3000/api/v1/users/register' \\\r\n--header 'Content-Type: application/json' \\\r\n--data-raw '{\r\n \"firstName\": \"Yash\", \r\n \"lastName\": \"Kumar\", \r\n \"email\": \"yash12345999@gmail.com\", \r\n \"password\": \"Yash123@#\" \r\n}"
},
"response": []
},
{
"name": "http://localhost:3000/api/v1/users/login",

```

```

"request": {
  "method": "POST",
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ],
  "body": {
    "mode": "raw",
    "raw": "{\r\n \"email\": \"yash12345999@gmail.com\", \r\n \"password\":\"Yash123@#\" \r\n}",
    "options": {
      "raw": {
        "language": "json"
      }
    }
  },
  "url": {
    "raw": "http://localhost:3000/api/v1/users/login",
    "protocol": "http",
    "host": [
      "localhost"
    ],
    "port": "3000",
    "path": [
      "api",
      "v1",
      "users",
      "login"
    ]
  },
  "description": "Generated from cURL: curl --location 'http://localhost:3000/api/v1/users/login' \r\n--header 'Content-Type: application/json' \r\n--data-raw '{\r\n \"email\": \"yash12345999@gmail.com\", \r\n \"password\":\"Yash123@#\" \r\n}'"
},
"response": []
}
]
}

```