

user.js ...\models

admin.js ...\router

.gitignore

flight.js

booking.js

auth_admin.js

auth_user.js

admin.js ...\models

adminVadilation

server > models > user.js > userSchema > email

```
1  const mongoose = require('mongoose');
2
3  const userSchema = new mongoose.Schema({
4    email: {
5      type: String,
6      required: true,
7      unique: true,
8      validate: {
9        validator: function (email) {
10          const emailRegex = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;
11          return emailRegex.test(email);
12        },
13        message: 'Invalid email format',
14      },
15    },
16    password: {
17      type: String,
18      required: true,
19      minlength: 6,
20    },
21  });
22
23  const User = mongoose.model('User', userSchema);
24  module.exports = User;
25
```

user.js ...models admin.js ...router .gitignore flight.js booking.js auth_admin.js auth_user.js admin.js ...models adminVadilation

server > models > flight.js > flightSchema

```
1  const mongoose = require('mongoose');
2
3  const flightSchema = new mongoose.Schema({
4    flightNumber: {
5      type: String,
6      required: true,
7      unique: true,
8    },
9    time: {
10     type: Date,
11     required: true,
12   },
13   seatsAvailable: {
14     type: Number,
15     default: 60,
16   },
17   destination: {
18     type: String,
19     required: true,
20   },
21 });
22
23 const Flight = mongoose.model('Flight', flightSchema);
24 module.exports = Flight;
25
```

server > models > booking.js > bookingSchema > flightNumber

```
1  const mongoose = require('mongoose');
2
3  const bookingSchema = new mongoose.Schema({
4    flightNumber: {
5      type: String,
6      required: true,
7    },
8    time: {
9      type: Date,
10     required: true,
11   },
12   passengerName: {
13     type: String,
14     required: true,
15   },
16   seatNumber: {
17     type: Number,
18     required: true,
19   },
20   passengerEmail: {
21     type: String,
22     required: true,
23   },
24 });
25
26 const Booking = mongoose.model('Booking', bookingSchema);
27 module.exports = Booking;
28
```

user.js ...models admin.js ...router .gitignore flight.js booking.js auth_admin.js auth_user.js admin.js ...models X adminVac

server > models > admin.js > adminSchema > email > validate > validator

```
1  const mongoose = require('mongoose');
2
3  const adminSchema = new mongoose.Schema({
4    email: {
5      type: String,
6      required: true,
7      unique: true,
8      validate: {
9        validator: function (email) {
10          const emailRegex = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;
11          return emailRegex.test(email);
12        },
13        message: 'Invalid email format',
14      },
15    },
16    password: {
17      type: String,
18      required: true,
19      minlength: 6,
20    },
21  });
22
23  const Admin = mongoose.model('Admin', adminSchema);
24  module.exports = Admin;
25
```