

## URL Shortener - Code and Explanation

### server.js

```
const express = require('express');           // Import express for web server
const mongoose = require('mongoose');         // Import mongoose for MongoDB
const cors = require('cors');                 // To allow API access from frontend
const shortRoutes = require('./routes/shortUrlRoutes'); // Import shortener routes

const app = express();
app.use(cors());                             // Enable CORS
app.use(express.json());                     // Parse JSON body
app.use('/', shortRoutes);                   // Use shortener routes

mongoose.connect('mongodb://127.0.0.1:27017/urlShortener') // Connect to DB
  .then(() => console.log("MongoDB connected"));

app.listen(5000, () => console.log("Server running on http://localhost:5000"));
```

### models/ShortUrl.js

```
const mongoose = require('mongoose');

const ShortUrlSchema = new mongoose.Schema({
  url: String,                                // Original long URL
  shortcode: { type: String, unique: true }, // Custom or generated code
  createdAt: { type: Date, default: Date.now },
  expiryAt: Date,                             // Expiry time
  clicks: [{ timestamp: Date, referrer: String, geo: String }] // Click logs
});

module.exports = mongoose.model('ShortUrl', ShortUrlSchema); // Export model
```

### routes/shortUrlRoutes.js

```
const express = require('express');
const router = express.Router();
const ShortUrl = require('../models/ShortUrl');
const { v4: uuidv4 } = require('uuid');      // UUID for generating shortcodes

// Create short URL
router.post('/shorturls', async (req, res) => {
  const { url, validity = 30, shortcode } = req.body;
  const code = shortcode || uuidv4().slice(0, 6);
  const expiry = new Date(Date.now() + validity * 60000);

  const exists = await ShortUrl.findOne({ shortcode: code });
  if (exists) return res.status(400).json({ error: "Shortcode already used" });

  const short = await ShortUrl.create({ url, shortcode: code, expiryAt: expiry });
  res.status(201).json({
    shortLink: `http://localhost:5000/shorturls/${code}`,
    expiry: expiry.toISOString()
  });
});
```

## URL Shortener - Code and Explanation

```
    });  
  });  
  
  // Redirect to original  
  router.get('/shorturls/:code', async (req, res) => {  
    const { code } = req.params;  
    const data = await ShortUrl.findOne({ shortcode: code });  
  
    if (!data) return res.status(404).json({ error: "Short URL not found" });  
    if (Date.now() > new Date(data.expiryAt)) return res.status(410).json({ error: "URL expired" });  
  
    data.clicks.push({ timestamp: new Date(), referrer: req.headers.referrer || 'unknown', geo: 'N/A' });  
    await data.save();  
  
    res.redirect(data.url);  
  });  
  
  // Stats route  
  router.get('/shorturls/stats/:code', async (req, res) => {  
    const { code } = req.params;  
    const data = await ShortUrl.findOne({ shortcode: code });  
    if (!data) return res.status(404).json({ error: "Short URL not found" });  
  
    res.json({  
      totalClicks: data.clicks.length,  
      originalUrl: data.url,  
      createdAt: data.createdAt,  
      expiryAt: data.expiryAt,  
      clicks: data.clicks  
    });  
  });  
  
  module.exports = router;
```