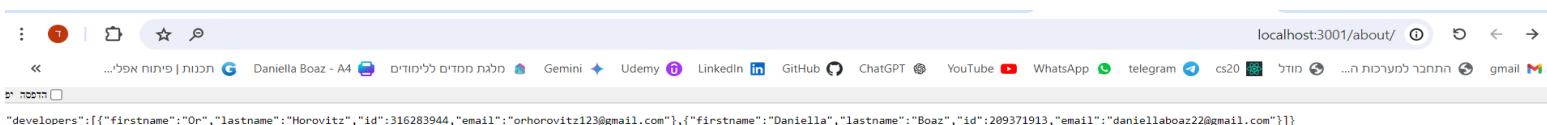


# Calories Manager REStful Web Services

Final Project in Asynchronous Server Side Course

Submitted by:

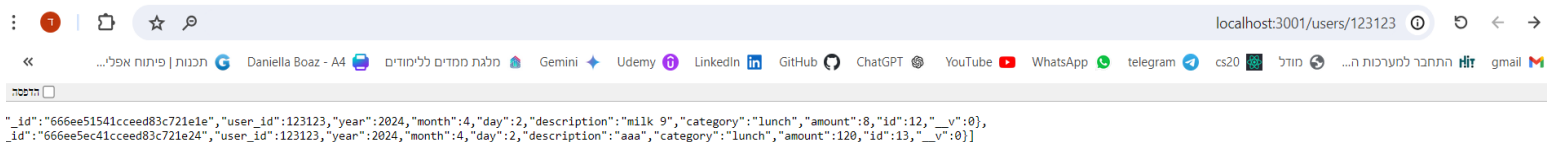
Name	ID	Email	Mobile Number
Daniella Boaz	209371913	Daniellaboaz22@gmail.com	0549000716
Or Horovitz	316283944	orhorovitz123@gmail.com	0504701376



```

{
  "developers": [
    {
      "firstname": "Or",
      "lastname": "Horovitz",
      "id": "316283944",
      "email": "orhorovitz123@gmail.com"
    },
    {
      "firstname": "Daniella",
      "lastname": "Boaz",
      "id": "209371913",
      "email": "daniellaboaz22@gmail.com"
    }
  ]
}

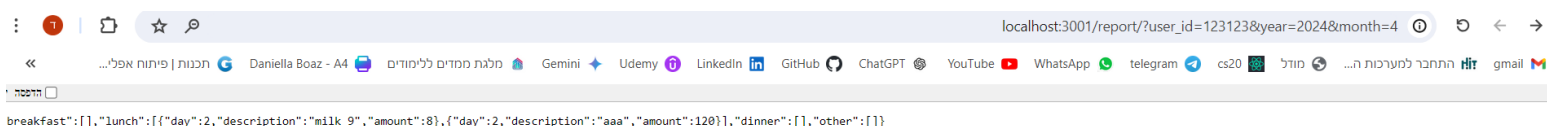
```



```

{
  "id": "666ee51541ccedd83c721e1e",
  "user_id": "123123",
  "year": "2024",
  "month": "4",
  "day": "2",
  "description": "milk 9",
  "category": "lunch",
  "amount": "8",
  "id": "12",
  "v": "0",
  "id": "666ee5ec41ccedd83c721e24",
  "user_id": "123123",
  "year": "2024",
  "month": "4",
  "day": "2",
  "description": "aaa",
  "category": "lunch",
  "amount": "120",
  "id": "13",
  "v": "0"
}

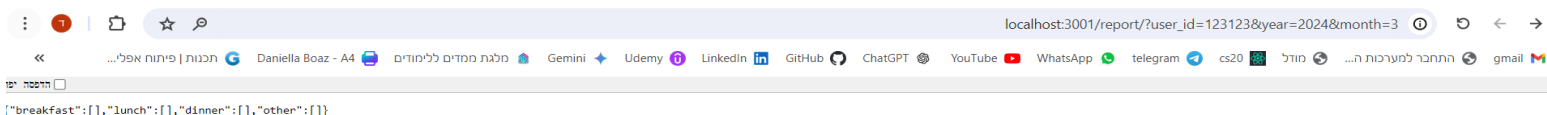
```



```

{
  "breakfast": [],
  "lunch": [
    {
      "day": "2",
      "description": "milk 9",
      "amount": "8"
    },
    {
      "day": "2",
      "description": "aaa",
      "amount": "120"
    }
  ],
  "dinner": [],
  "other": []
}

```



```

{
  "breakfast": [],
  "lunch": [],
  "dinner": [],
  "other": []
}

```

Video Link: <https://youtu.be/pH7tSUefyQo>

GitHub Link: <https://github.com/daniellaboaz/REStful-Web-Services.git>

Web Link: <https://restful-web-services.onrender.com>

**table of contents:**

controllers/

[caloriesController.js.....3-4](#)

models/

[calories.js.....5](#)

routes/

[about.js.....6](#)[addcalories.js.....7](#)[index.js.....8](#)[report.js.....9](#)[users.js.....10](#)

views/

[index.pug.....11](#)[app.js.....12](#)

## controllers/caloriesController.js:

```
// - Or Horovitz 316283944, Daniella Boaz 209371913
// - controllers/caloriesController

const caloriesModel = require('../models/calories');

// Function to create a new calories entry
async function createCalories(caloriesData) {
  try {
    const calories = new caloriesModel(caloriesData);
    await calories.save();
    return calories;
  } catch (error) {
    console.error('Error creating calories entry:', error);
    throw error;
  }
}

// Get report by user_id
async function getCaloriesByUserId(user_id) {
  try {
    const calories = await caloriesModel.find({ user_id });
    return calories;
  } catch (error) {
    console.error('Error getting calories entries by user_id:', error);
    throw error;
  }
}

// Generate detailed calories report for specific month, year, and user_id
async function generateCaloriesReport(user_id, month, year) {
  try {
    const calories = await caloriesModel.find({
      user_id,
      month,
      year
    });

    // Initialize an object to store the report
    const report = {
      breakfast: [],
      lunch: [],
      dinner: [],
      other: []
    };

    // Populate the report object based on category
    calories.forEach(calorie => {
      switch (calorie.category.toLowerCase()) {
        case 'breakfast':
          report.breakfast.push({
            day: calorie.day,
            description: calorie.description,
            amount: calorie.amount
          });
          break;
        case 'lunch':
          report.lunch.push({
            day: calorie.day,
            description: calorie.description,
            amount: calorie.amount
          });
      }
    });
  }
}
```

```
        break;
      case 'dinner':
        report.dinner.push({
          day: calorie.day,
          description: calorie.description,
          amount: calorie.amount
        });
        break;
      default:
        report.other.push({
          day: calorie.day,
          description: calorie.description,
          amount: calorie.amount
        });
    }
  });

  // Ensure all categories (even empty ones) are included in the report
  ['breakfast', 'lunch', 'dinner', 'other'].forEach(category => {
    if (!report[category].length) {
      report[category] = [];
    }
  });

  return report;
} catch (error) {
  console.error('Error generating calories report:', error);
  throw error;
}
}

module.exports = { createCalories ,getCaloriesByUserId ,generateCaloriesReport };
```

## models/calories.js:

```
// - Or Horovitz 316283944, Daniella Boaz 209371913
// - models/calories

const mongoose = require('mongoose');

// Counter Schema for tracking sequence numbers
const counterSchema = new mongoose.Schema({
  _id: { type: String, required: true },
  seq: { type: Number, default: 0 }
});

const Counter = mongoose.model('Counter', counterSchema);

// Calories Schema definition
const caloriesSchema = new mongoose.Schema({
  id: Number,
  user_id: Number,
  year: Number,
  month: Number,
  day: Number,
  description: String,
  category: String,
  amount: Number,
});

// Pre-save hook to generate the id before saving to the database
caloriesSchema.pre('save', async function (next) {
  if (this.isNew) {
    // Only generate new id if the document is new
    try {
      const counter = await Counter.findByIdAndUpdate(
        { _id: 'caloriesId' },
        { $inc: { seq: 1 } },
        { new: true, upsert: true } // return new document, create if not exists
      );
      // Assign the generated sequence number to 'id' field
      this.id = counter.seq;
      next();
    } catch (error) {
      next(error);
    }
  } else {
    next();
  }
});

const Calories = mongoose.model('Calories', caloriesSchema);

module.exports = Calories;
```

## routes/about.js:

```
// Daniella Boaz 209371913, Or Horovitz 316283944
// routes/about.js

const express = require('express');
const router = express.Router();

// Define array of developers
const developers = [
  { firstname: 'Or', lastname: 'Horovitz', id: 316283944,
    email: 'orhorovitz123@gmail.com' },
  { firstname: 'Daniella', lastname: 'Boaz', id: 209371913,
    email: 'daniellaboaz22@gmail.com' }
];

// Route to get JSON array of developers
router.get('/', (req, res) => {
  res.json({ developers }); // Return JSON object
});

module.exports = router;
```

## routes/addcalories.js:

```
// - Or Horovitz 316283944, Daniella Boaz 209371913
// routes/addcalories.js

const express = require('express');
const router = express.Router();
const caloriesController = require('../controllers/caloriesController');

router.get('/', function(req, res, next) {
  res.render('addcalories', { college: 'addcalories' });
});

// POST request to handle submission from server
router.post('/', async (req, res) => {
  try {
    const { user_id, year, month, day, description, category, amount } = req.body;

    // Create calories entry
    const caloriesData = {
      user_id: parseInt(user_id),
      year: parseInt(year),
      month: parseInt(month),
      day: parseInt(day),
      description,
      category,
      amount: parseInt(amount)
    };

    const newCaloriesEntry = await caloriesController.createCalories(caloriesData);

    // Check if newCaloriesEntry is false or null and throw an error if it is.
    if (!newCaloriesEntry) {
      throw new Error('Failed to create calories entry');
    }
    // Respond with the newly created entry
    res.status(201).json(newCaloriesEntry);
  } catch (error) {
    console.error('Error adding calories:', error);
    // Return error message as JSON response
    res.status(500).json({ error: error.message });
  }
});

module.exports = router;
```

## routes/index.js:

```
// - Or Horovitz 316283944, Daniella Boaz 209371913
// routes/index.js

const express = require('express');
const router = express.Router();

// GET home page
router.get('/', function(req, res, next) {
  res.render('index', { title: 'Home' });
});

module.exports = router;
```



## routes/report.js:

```
// - Or Horovitz 316283944, Daniella Boaz 209371913
// routes/report.js

const express = require('express');
const router = express.Router();
const caloriesController = require('../controllers/caloriesController');

// GET detailed report for specific month, year, and user_id
router.get('/', async function(req, res) {
  const { user_id, month, year } = req.query;

  // check if the user insert all the parameters
  if (!user_id || !month || !year) {
    return res.status(400)
      .json({ error: 'Missing parameters: user_id, month, and year are required.' });
  }

  try {
    const report = await caloriesController
      .generateCaloriesReport(user_id, parseInt(month), parseInt(year));
    res.json(report); // Respond with JSON data
  } catch (error) {
    console.error('Error generating report:', error);
    res.status(500).json({ error: 'Internal server error' });
  }
});

module.exports = router;
```

## routes/users.js:

```
// - Or Horovitz 316283944, Daniella Boaz 209371913
// routes/user.js

const express = require('express');
const router = express.Router();
const caloriesController = require('../controllers/caloriesController');

// GET specific user report
router.get('/:user_id', async function(req, res) {
  try {
    const user_id = req.params.user_id;
    const calories =
      await caloriesController.getCaloriesByUserId(user_id);
    res.json(calories); // Respond with JSON data
  } catch (error) {
    console.error('Error getting calories:', error);
    res.status(500).json({ error: 'Internal server error' });
  }
});

// GET products listing
router.get('/', function(req, res, next) {
  res.send('this is the users page, Please enter user id');
});

module.exports = router;
```

## views/index.pug:

```
//- Or Horovitz 316283944, Daniella Boaz 209371913
//- views/index.pug
extends layout

block content
  .container
    h1 Welcome to Home
    p This is the home page of our website.
    p From here, you can navigate to other sections of the site to learn more about our services,
    p add your daily calorie intake, generate reports, and view user information.
```

## app.js:

```
// - Or Horovitz 316283944, Daniella Boaz 209371913
// - app.js
const express = require('express');
const bodyParser = require('body-parser');
const { MongoClient } = require('mongodb');
const app = express();
const path = require('path');
const cookieParser = require('cookie-parser');
const logger = require('morgan');
const mongoose = require('mongoose');

// Import routes
const indexRouter = require('./routes/index');
const usersRouter = require('./routes/users');
const aboutRouter = require('./routes/about');
const addRouter = require('./routes/addcalories');
const reportRouter = require('./routes/report');

// Middleware
app.use(logger('dev'));
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));

// Set the view engine to Pug
app.set('view engine', 'pug');
app.set('views', path.join(__dirname, 'views'));

// MongoDB Connection
const uri = process.env.MONGODB_URI
  || 'mongodb+srv://orhorovitz123:NLx1tDZQ8E9mtFJf@cluster0.t0r5fi2.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0';

// try to connect to MongoDB
mongoose.connect(uri, {
  serverSelectionTimeoutMS: 5000,
  socketTimeoutMS: 45000,
}).then(() => {
  console.log('MongoDB connected...');
}).catch(err => {
  console.error('MongoDB connection error:', err);
  process.exit(1);
});

// Routes
app.use('/', indexRouter);
app.use('/users', usersRouter);
app.use('/about', aboutRouter);
app.use('/addcalories', addRouter);
app.use('/report', reportRouter);

// Error Handling for incorrect URL
app.use((req, res, next) => {
  res.status(404).send('URL not correct');
});

// Start the server and listen on the specified port
const PORT = process.env.PORT || 3001;
app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
module.exports = app;
```