Express Generator installs automatically several packages as dependencies: **cookie-parser**, **jade, mongoose**, etc.

Part 1:

Research these packages. Pick 3 of them (aside from Express) and look up their official online documentation pages. Explain briefly what each package is used for overall, then choose one specific part of the package's API and explain in your own words what it does, and how it is used. Add a code snippet on how to use it. Pick a part of each API that has not already been described by another student. Include the link to the online documentation you used, along with links to any articles you read to supplement the online documentation. Note: There is one package installed by express-generator that is deprecated. If you choose to use this one for one of your answers, then include information about the deprecation and suggested updates.

cookie-parser - https://www.npmjs.com/package/cookie-parser

Parse `Cookie` header and populate `req.cookies` with an object keyed by the cookie names. Optionally you may enable signed cookie support by passing a `secret` string, which assigns `req.secret` so it may be used by other middleware.

The middleware will parse the `Cookie` header on the request and expose the cookie data as the property `req.cookies.`

```
var cookieParser = require('cookie-parser')
app.use(cookieParser())

app.get('/', function (req, res) {
  console.log('Cookies: ', req.cookies)
```

jade

This package has been deprecated

Author message:

Jade has been renamed to pug, please install the latest version of pug instead of jade This project was formerly known as "Jade." However, it was revealed that "Jade" is a registered trademark, and as a result a rename is needed

pug - https://www.npmjs.com/package/pug

Pug is a high performance template engine heavily influenced by **Haml** and implemented with JavaScript for **Node.js** and browsers.

The general rendering process of Pug is simple. `pug.compile()` will compile the Pug source code into a JavaScript function that takes a data object (called “`locals`”) as an argument. Call that resultant function with your data, and *voilà!*, it will return a string of HTML rendered with your data.

The compiled function can be re-used, and called with different sets of data.

```
//- template.pug
p #{name}'s Pug source code!
```

```
//-----------------------------------------------------
const pug = require('pug');
// Compile the source code
const compiledFunction = pug.compileFile('template.pug');

// Render a set of data
console.log(compiledFunction({
  name: 'Timothy'
}));
// "<p>Timothy's Pug source code!</p>"

// Render another set of data
console.log(compiledFunction({
  name: 'Forbes'
}));
// "<p>Forbes's Pug source code!</p>"
```

mongoose

Mongoose is a **MongoDB** object modeling tool designed to work in an asynchronous environment. Mongoose supports both promises and callbacks.

Defining your schema

Everything in Mongoose starts with a Schema. Each schema maps to a MongoDB collection and defines the shape of the documents within that collection.

```
import mongoose from 'mongoose';
const { Schema } = mongoose;

const blogSchema = new Schema({
  title:  String, // String is shorthand for {type: String}
  author: String,
  body:   String,
  comments: [{ body: String, date: Date }],
  date: { type: Date, default: Date.now },
  hidden: Boolean,
  meta: {
    votes: Number,
    favs:  Number
  }
});
```

Part 2:

What are some other packages that could be useful to use with Express? Research this question and provide your answer below with links to articles and sample code.

cors - https://github.com/expressjs/cors

CORS is a node.js package for providing a Connect/Express middleware that can be used to enable CORS with various options.

Simple Usage (Enable *All* CORS Requests)

```
var express = require('express')
var cors = require('cors')
var app = express()

app.use(cors())

app.get('/products/:id', function (req, res, next) {
  res.json({msg: 'This is CORS-enabled for all origins!'})
})

app.listen(80, function () {
  console.log('CORS-enabled web server listening on port 80')
})
```

Morgan - https://github.com/expressjs/morgan

HTTP request logger middleware for node.js

```
var morgan = require('morgan')
```