Introduction to Backend.

Internet Basics:

① url query is submitted to your ISP, within ISP, the DNS takes the Domain name and turns it into an IP address.

② A request is sent to the desired IP address via HTTP, and the request finds the fastest path to the server with the specified IP.

③ The requested server figures out exactly what we are asking for, and the server builds us the right content, often pulling information from database. The server responds with any combination of HTML, CSS, and JavaScript.

Static vs Dynamic Website Design

① Static site is one usually written in plain HTML/CSS/JavaScript and what is in the code of the page is what is displayed to the user.

② Dynamic site is one that is written using a server-side scripting language such as php, ASP, JSP, or coldfusion. The site content is called in by the scripting language from other files or from a database depending on actions taken by the user.

Advantages/Disadvantages:

Static sites: Advantages: flexibility: each page can be designed differently.

Disadvantages: ① update the content is costly.
② scalability is a big issue: have to construct each individual page.

Dynamic sites:

Advantages: ① well-organised and structured way to create pages by connecting to database.
② update the content is cheap by creating a content management system with connecting to a database.

Disadvantages: ① The design is more fixed than a static site.
② Cost of building a dynamic web pages is much higher than static website.

Command Line.

make a new file: touch newfile.txt
make a folder: mkdir
ls, cd, rm,
to delete a folder: rm -rf folder

NodeJS (JavaScript for backend).

To run a file in node: node file.js

NPM: the package manager for JavaScript.

Installing and using packages:

① npm install package-name // install the package at first.
② require("package-name"); //=> return the function from the package.

To find more useful packages, go to NPM website.

Introduction to Express (web framework for Node.js)

Difference between frameworks and library: inversion of control.

i.e: when you call a library, you are in control.
But with a framework, all the control flow is already in the framework, and there's just a bunch of predefined white spots that you can fill out with your code.

Install: npm install express (--save) => save the package as dependencies

Then: var express = require("express");
var app = express();

Routing methods of Express: (How to respond to client requests).

A route method is derived from one of the HTTP methods, and is attached to an instance of the express class.

methods: get, post, put, head, delete, options, ..., etc.

i.e:
```
app.get('/about', function(req, res){
    res.send("data");
});
```

```
app.listen(process.env.PORT, process.env.IP);
```

The package.json file:

Use "npm init" to create a new package.json file.

"name" and "version" are the two most important things in package.json file.

Changes to the package should also come along with changes to the version.

Route parameters:

"/:pattern"

① Route paths can be strings, string patterns or regular expressions.

?: 0 or 1 occurence of the preceding element.

*: any expression

+: 1 or more occurences of the preceding elements.

( ): groups elements together.

In cloud9, an easy way to open a file : c9 file.name

EJS : a simple templating language that lets you generate HTML markup with plain JavaScript. (helps to build dynamic html.

install EJS : npm install ejs

Then we can send .ejs file using : .render(".ejs", template variable.

In .ejs, you can define any templates using tag <%= .... %>,

Then we can pass the value of templates using . render("file.ejs", {temp: val})

EJS Control flow:

Useful tags:

<% %> : scriptlet tag, for control flow, no output.
<%= %> : output the value into the template (HTML escaped)
<%# ... %> : comment tag, no execution, no output
<%- %> : output the unescaped value into the template.

i.e: <% if (var === "hello") { %> } add <% %> at each line.
...
<% } %>

To add style in ejs file, using : <style> </style> in .ejs
or we use link : <link rel="stylesheet" href="app.css">

To load/use a directory, use : app.use(express.static("folder_name");

To avoid type .ejs postfix each time, we can use:

app.set("view engine", "ejs"); then all .ejs files don't have to include .ejs

In order to read HTTP post data, we have to use "body-parser", which is a piece of express middleware that reads a form's input and stores it as a javascript object accessible through req.body. npm install body-parser --save.