Recitation #2

17-356/17-766

TAs

- **Mehul Agarwal**
    - email: mehula@andrew.cmu.edu
    - office hours: TBD

- **Rohit Shreenivas**
    - email: rshreeni@andrew.cmu.edu
    - office hours: TBD

Full-stack Development

Different levels of the stack:

- Backend
- Frontend
- Database
- Deployment
- Testing

and more

Full-stack Development

Different levels of the stack:

- **Backend** → Today’s topic
- Frontend
- Database
- Deployment
- Testing

and more

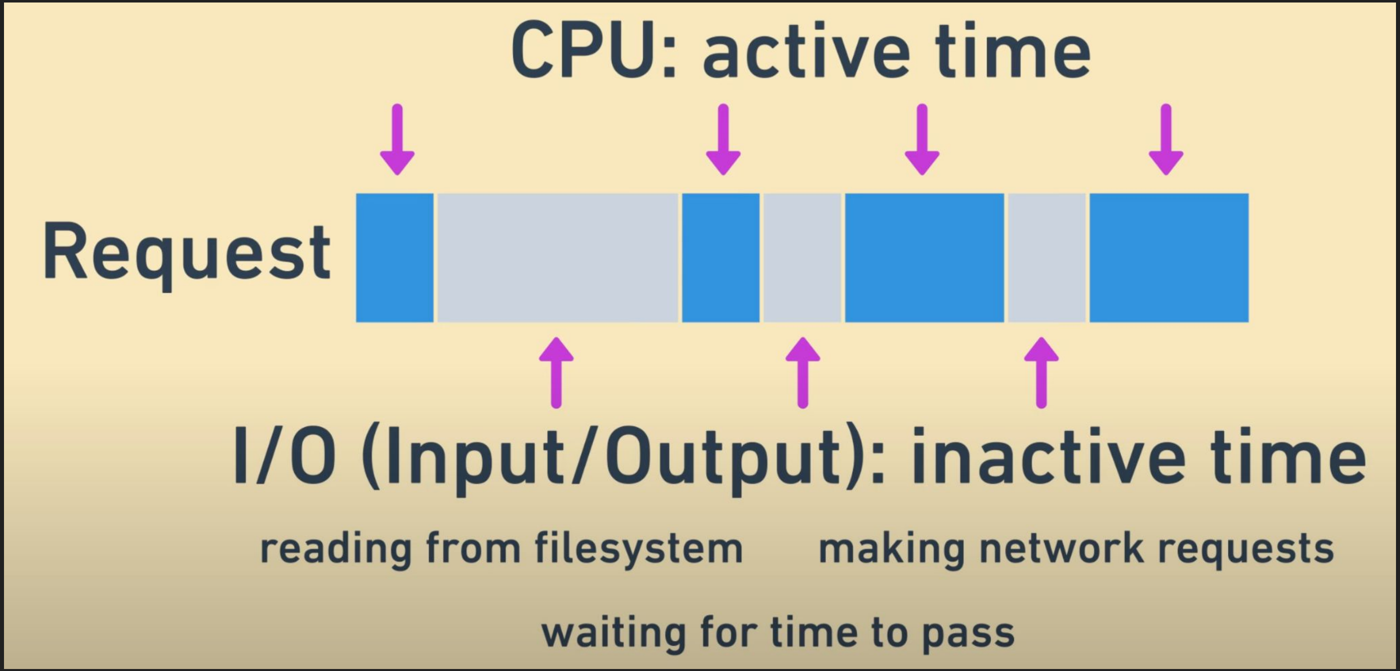
BACKEND DEVELOPMENT USING NODE.JS

What is Node?

- Back-end JavaScript runtime environment, runs on the V8 JavaScript Engine
- Used for server-side scripting
- Has an event-driven architecture capable of asynchronous(non-blocking) I/O.
- Used to build fast, powerful and scalable web applications

Asynchronous or Non-blocking I/O

Every request consists of:



Asynchronous or Non-blocking I/O

- Non-blocking I/O allows a thread to suspend a request while it’s performing I/O to go work on a different request
- Runs on a single-thread event loop, using non-blocking I/O calls, allowing it to support tens of thousands of concurrent connections without incurring the cost of thread context switching
- Concurrency (in Node) refers to the Event Loop’s capacity to execute Javascript “callback” functions after completing other work

Node js async options

Callbacks: Functions passed to another functions

Promises (and promise chains): Structured callbacks

Generators: Functions which can be exited/paused and later re-entered

Async/Await: Combining generators and promises

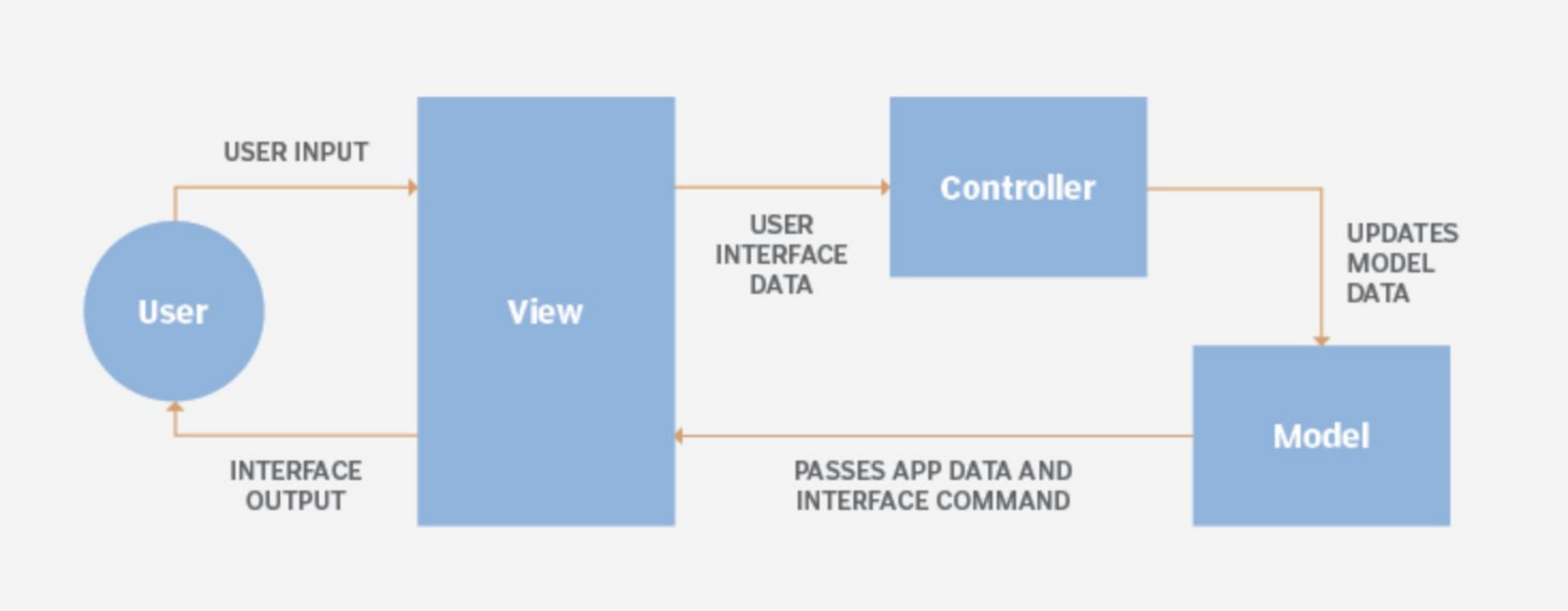
Node Package Manager (npm)

Consists of:

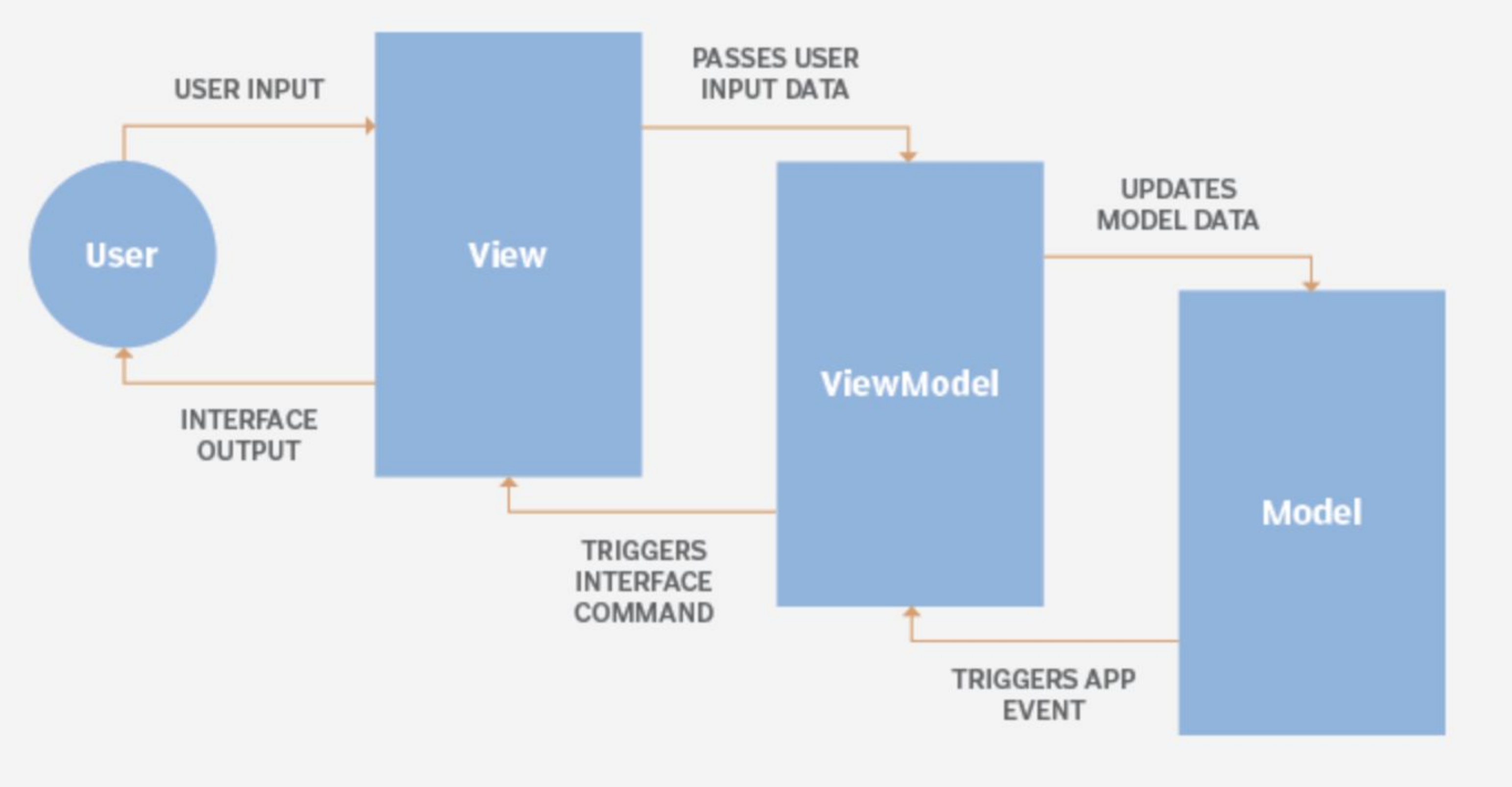
- npm client - command line tool
- npm registry - an online database of public and paid-for private packages

The largest ecosystem of open-source libraries in the world

Model View Controller (MVC)



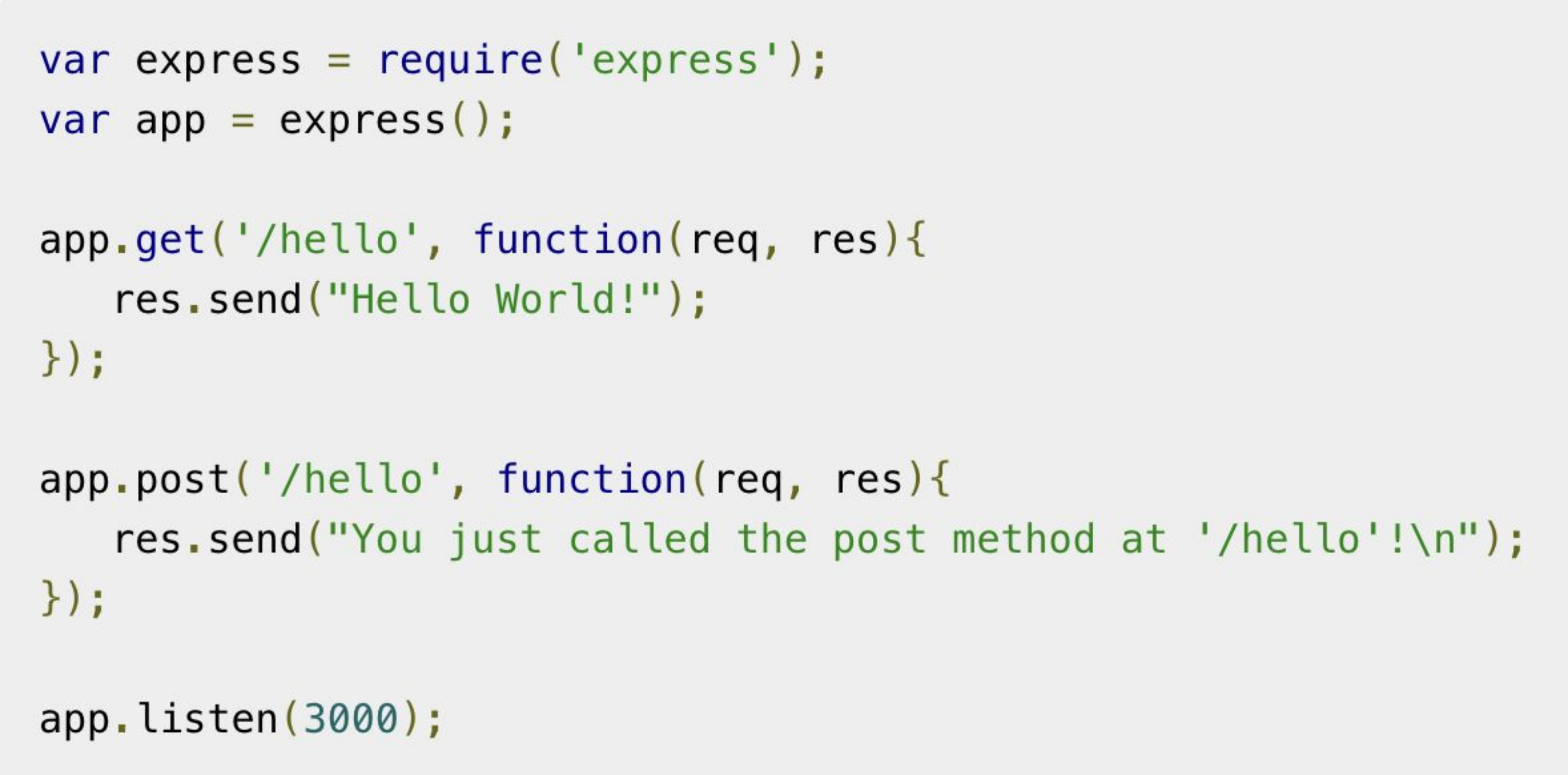
Model View Viewmodel (MVVM)



A SIMPLE TO-DO APP USING NODE.JS

STEPS:

1. ***mkdir*** your project folder (todo-app) and ***cd*** into it
2. ***npm init***
3. ***npm install express --save***
4. ***npm install body-parser --save***
5. Write express code, as seen in the reference solution
6. ***node index.js*** (to start the server)
7. (optional) Play with the server, send it post requests



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

app.get('/hello', function(req, res){
   res.send("Hello World!");
});

app.post('/hello', function(req, res){
   res.send("You just called the post method at '/hello'!\n");
});

app.listen(3000);
```