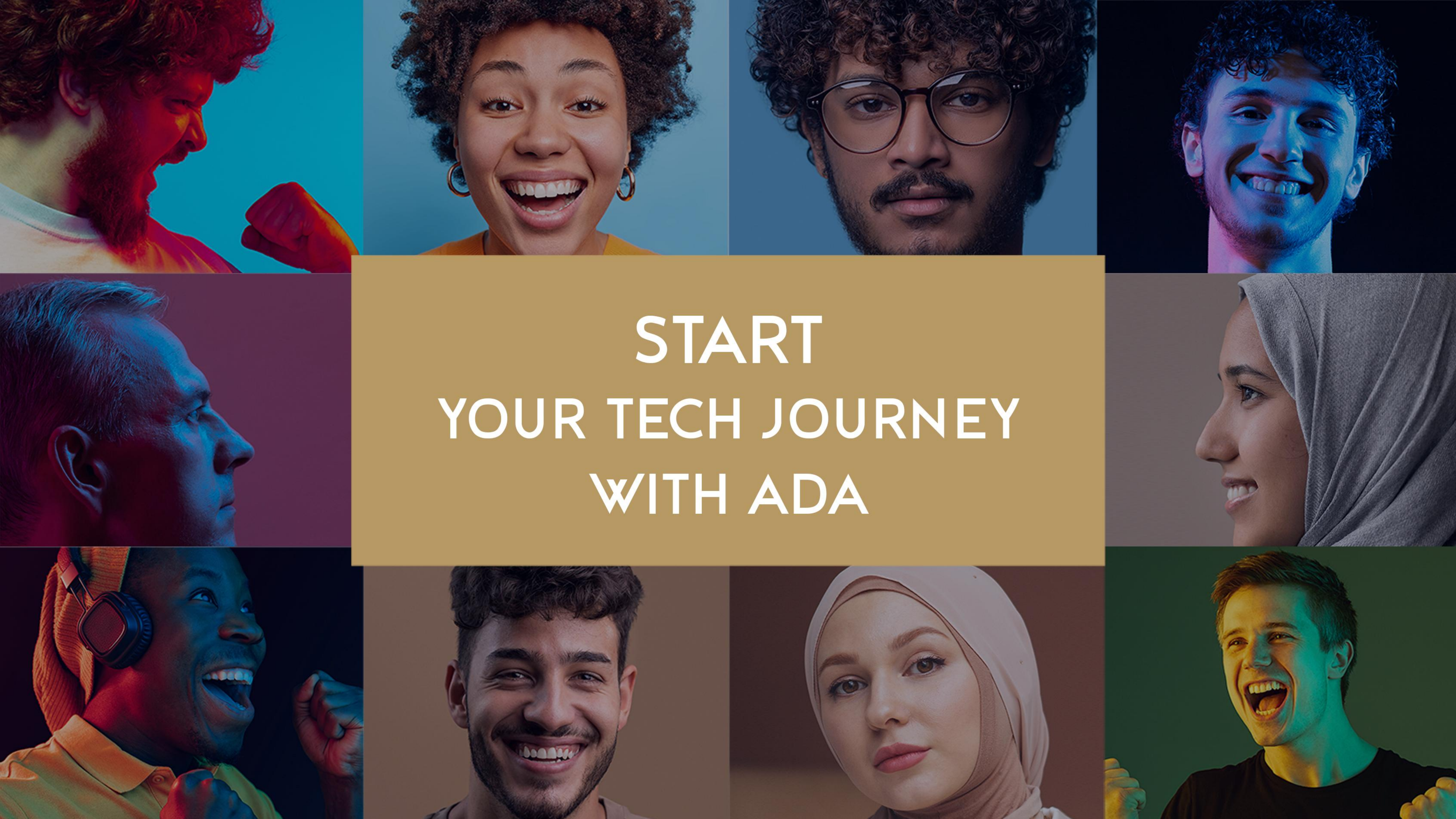
ACADEMY OF DIGITAL ARTS

EGYPT

Google cisco Adobe IBM Microsoft CompTIA



START YOUR TECH JOURNEY WITH ADA

Node.js Session 1 - Student Task

Academy of Digital Arts Egypt - Back-End Development Course

Objective:

By completing this task, you will:

- Set up Node.js environment and verify installation correctly
- Create and use CommonJS modules with require() and module.exports
- Understand the difference between synchronous and asynchronous execution
- Implement callback functions following the error-first pattern
- Use built-in Node.js modules like path and os for system information
- Build a complete calculator system demonstrating Node.js fundamentals

Node.js Session 1 - Student Task

Academy of Digital Arts Egypt - Back-End Development Course

Step 1 – Set Up Node.js Environment

- Install Node.js LTS version from nodejs.org
- Verify installation using node --version and npm --version commands
- Create a project folder and navigate to it in terminal
- Test Node.js by running a simple "Hello World" program

Node.js Session 1 - Student Task

Academy of Digital Arts Egypt - Back-End Development Course

Step 2 – Create Calculator Modules

- Build math.js module with basic operations (add, subtract, multiply, divide)
- Create logger.js module for displaying messages with timestamps
- Export functions using module.exports syntax
- Import modules using require() in a main application file

Node.js Session 1 - Student Task

Academy of Digital Arts Egypt - Back-End Development Course

Step 3 – Implement Async Operations with Callbacks

- Create timer.js module with delay function using setTimeout
- Implement error-first callback pattern (callback(error, result))
- Build async calculation functions that simulate processing time
- Handle both success and error cases in callback functions

Node.js Session 1 - Student Task

Academy of Digital Arts Egypt - Back-End Development Course

Step 4 – Use Built-in Modules and System Info

- ➢ Use path module to work with file paths and get current directory
- ➢ Use os module to display system information (platform, architecture, memory)
- ➢ Create a system info module that exports computer details
- ➢ Combine all modules in a main application demonstrating Node.js concepts

Node.js Session 1 - Student Task

Academy of Digital Arts Egypt - Back-End Development Course

Requirements

Tools:

- Computer with internet connection for Node.js download
- Text editor (VS Code recommended)
- Terminal or Command Prompt access
- Node.js LTS version installed

Node.js Session 1 - Student Task

Academy of Digital Arts Egypt - Back-End Development Course

Reminder

- Use ./ prefix when requiring local modules (require('./math’))
- Always handle errors first in callback functions: if (error) { ... }
- Export objects with module.exports = { function1, function2 }
- Use setTimeout() to simulate async operations without file I/O
- Test your modules by running: node filename.js
- Remember Node.js is single-threaded but non-blocking for async operations

THANK YOU

ADAEGY