

HỘI THẢO CÔNG NGHỆ NODEJS CẮT CÁNH CÙNG THỜI ĐẠI SỐ

 **THỜI GIAN**
08:00 SÁNG

 **THỨ BẢY**
28 THÁNG 05

 **HỘI TRƯỜNG KHU B**
TRƯỜNG ĐH SƯ PHẠM KỸ THUẬT

DIỄN GIẢ

NGÔ THANH TÙNG
(SOFTECH & APTECH EDUCATION)

ĐỊA CHỈ: 48 CAO THẮNG, Q. HẢI CHÂU, TP. ĐÀ NẴNG



Nội dung trình bày



- ▶ **Nodejs** là gì?
- ▶ Tại sao lại dùng **nodejs**?
- ▶ Bắt đầu với **nodejs** như thế nào?
- ▶ Ứng dụng của **nodejs** trong thực tế.
- ▶ Q & A

NodeJS là gì?

- ▶ **NodeJS** là mã nguồn mở chạy trên môi trường **V8** JavaScript runtime (một trình thông dịch JavaScript chạy cực nhanh trên trình duyệt Chrome).
- ▶ **Learn Once, Write Anywhere**: NodeJS giúp các nhà phát triển xây dựng các ứng dụng một cách đơn giản và dễ dàng mở rộng.

Điểm mạnh của NodeJS

- ▶ IO hướng sự kiện **bất đồng bộ** giúp **xử lý nhiều yêu cầu đồng thời**.
- ▶ Đáp ứng được những yêu cầu về **thời gian thực**.
- ▶ Có **tốc độ cực rất nhanh**, đáp ứng được nhu cầu sử dụng của khách **truy cập 'khổng lồ' trong thời gian ngắn**.
- ▶ Dùng JavaScript **cho cả Client & Server**, một ngôn ngữ lập trình rất dễ học.
- ▶ **Nhiều modules rất mạnh mẽ** và vẫn đang tiếp tục phát triển rất nhanh.
- ▶ Có một **cộng đồng lớn mạnh**, có nhiều mã nguồn mở được chia sẻ rộng rãi.
- ▶ **Tương thích với nhiều** thiết bị, nhiều hệ điều hành như MacOS, Window, Linux,

Điểm yếu của NodeJS

- ▶ **NodeJS không phù hợp với các tác vụ đòi hỏi nhiều CPU** mà chỉ phù hợp với những I/O như máy chủ web.
- ▶ Nếu bạn có một web hosting dùng chung, sẽ rất khó khăn nếu bạn tải lên một ứng dụng NodeJS. VPS và Dedicated server là một sự lựa chọn tốt hơn nhiều.

NodeJS rất dễ học và ứng dụng thực tế

As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications. In the following "hello world" example, many connections can be handled concurrently. Upon each connection, the callback is fired, but if there is no work to be done, Node.js will sleep.

```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```



VS



JavaScript vs TypeScript

Great for beginners



For high salary



Loved by programmers



For web development



For enterprise projects



Ứng dụng của node.js

Microsoft®

mozilla

ebay

Google

PayPal®

Linked **in**

YAHOO!®

Bắt đầu với NodeJS như thế nào?

Learn the Prerequisites

Learn the basics of Javascript

JavaScript

Event Emitters

Callbacks

REST

Buffers

Module system

Process Object

NPM

Learn the basics of npm
semantic versioning

Version, scripts, repository and other Properties

Node.js Basics

Template Literals

Multi-Line Strings

Arrow Functions

Enhanced Object literals

ECMAScript

GIT - Version Control (VSTS, GitHub, GitLab)

HTTP/HTTPS protocol

Learn to search for solutions

Data Structures and Algorithms

Clean code

General Development Skills

Angular (Google)

Dễ dùng, phổ biến

React (Facebook)

Dễ dùng, phổ biến

**WEB
FRONT-END**

React Native

Mobile cross-platform iOS and Android

Vue

Dễ dùng, phổ biến

Khác

Alpinejs, Svelte, Polymer, Ember

START WITH WEB FRONT-END

Koa.js
Mới, nhanh, gọn

Sails.js
Realtime

WEB FRAMEWORKS

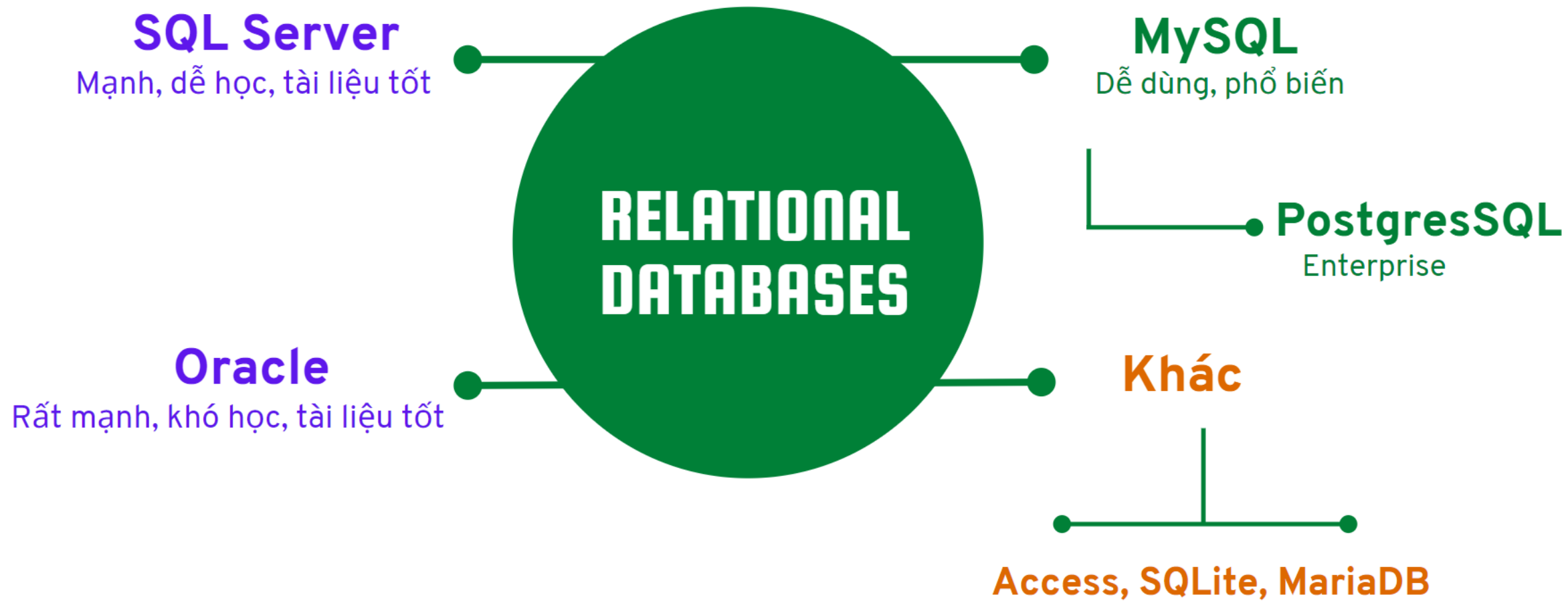
Express.js
Dễ dùng, phổ biến

Nest.js
Enterprise, Full technologies

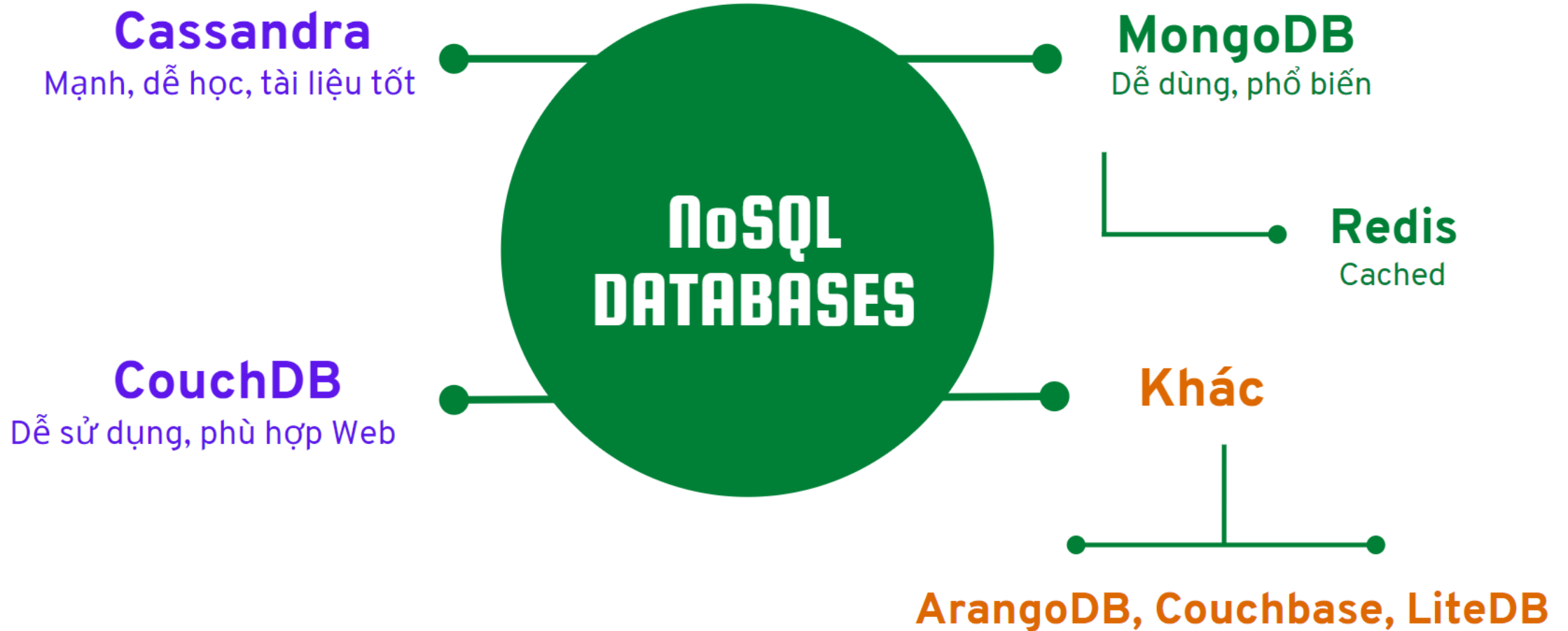
Khác

Loopback.js, Meteor.js, Adonis.js

START WITH WEB FRAMEWORKS



START WITH DATABASES



START WITH DATABASES

AWS DynamoDB

Mạnh, phổ biến, Enterprise

Firebase

Dễ dùng, phổ biến

**Cloud
DATABASES**

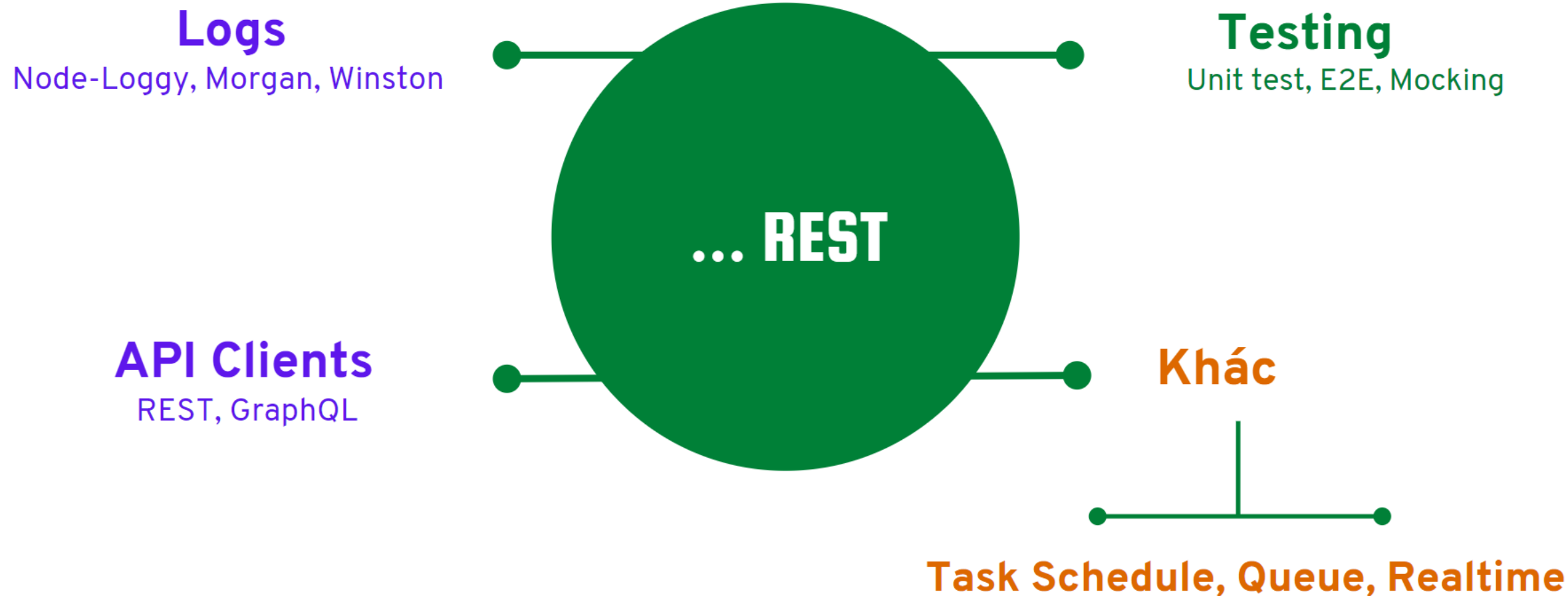
Azure CosmosDB

Mạnh, phổ biến, Enterprise

Khác

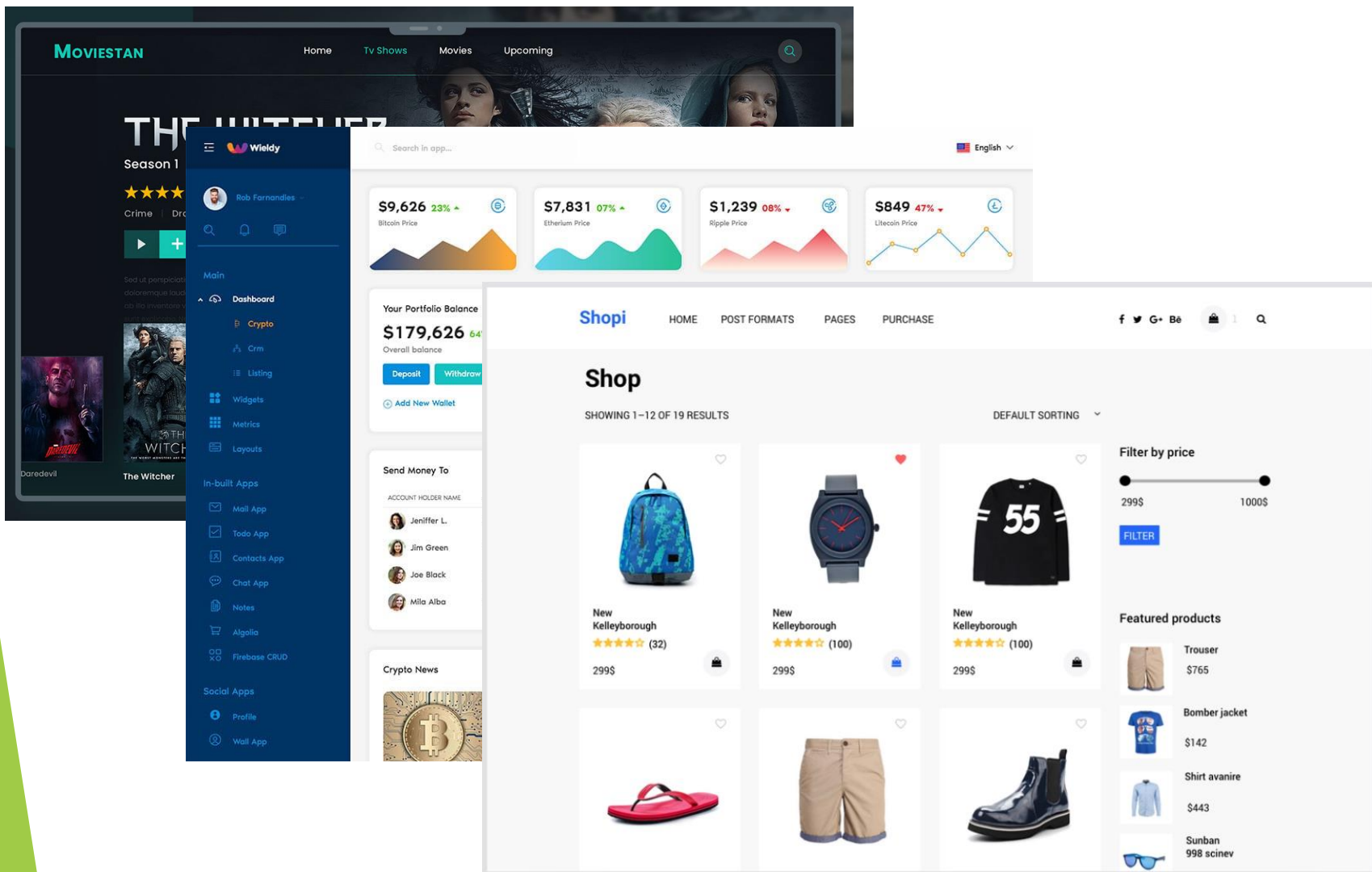
FaunaDB, RavenDB

START WITH DATABASES



... REST, BUT IMPORTANT

Nodejs với Web (Expressjs)



Node.js với Realtime



Fauget



Ngô Thanh Tùng

"We bend so we don't break."

Now

2



Aptech

"We change lives."

2 Min



Trường ĐH SPKT

"48 Cao Thắng, Đà Nẵng."

1 Hour

3



Aaron Loeb

"Allow yourself joy."

5 Hours



Lorna Alvarado

"Keep it simple."

22 Hours

4



Olivia Wilson

"You're wonderful."

Yesterday



Yael Amari

"You got this."

2 Weeks ago

1



Chats



Status

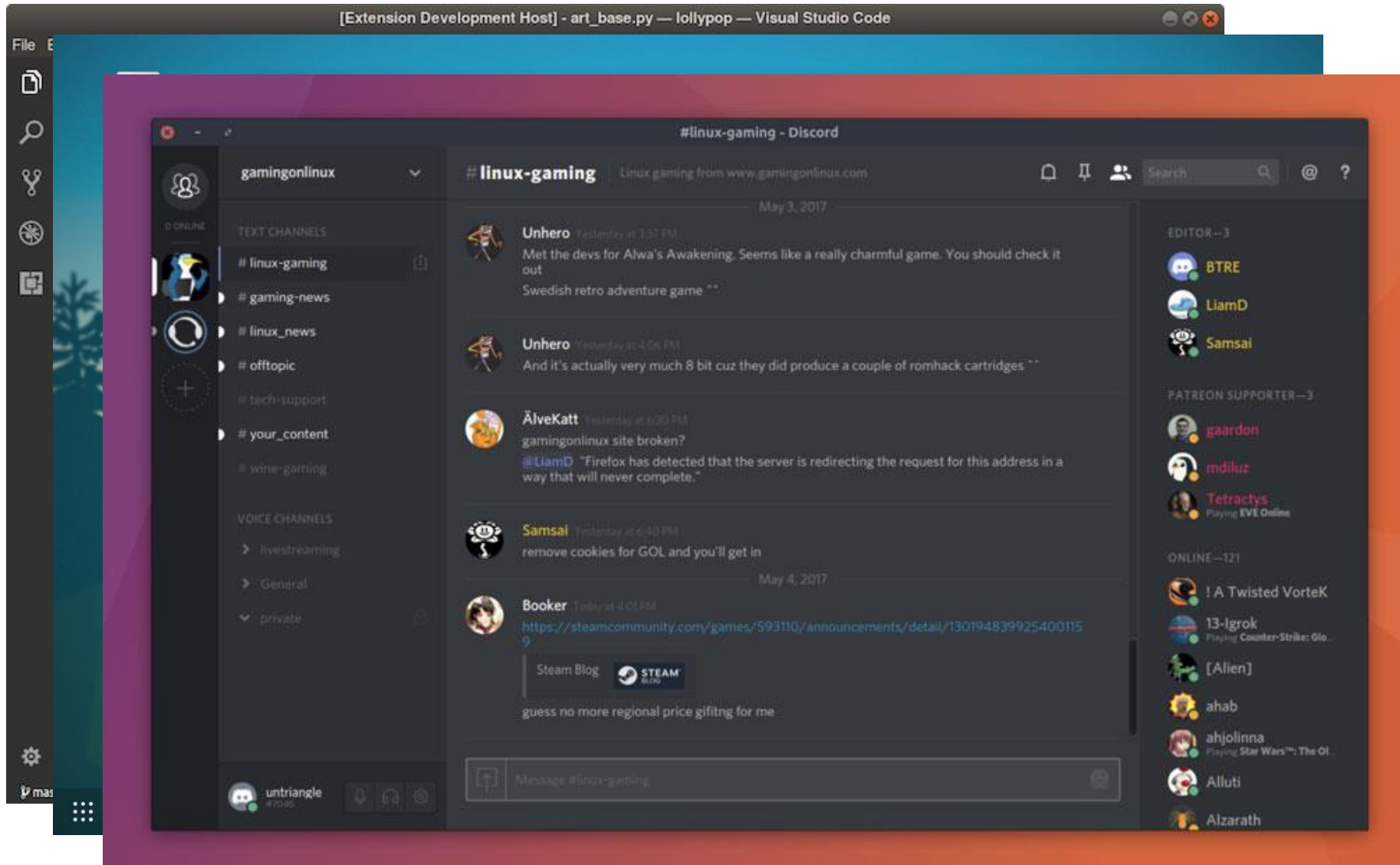


Calls



Profile

Node.js với Desktop Apps (Electron)

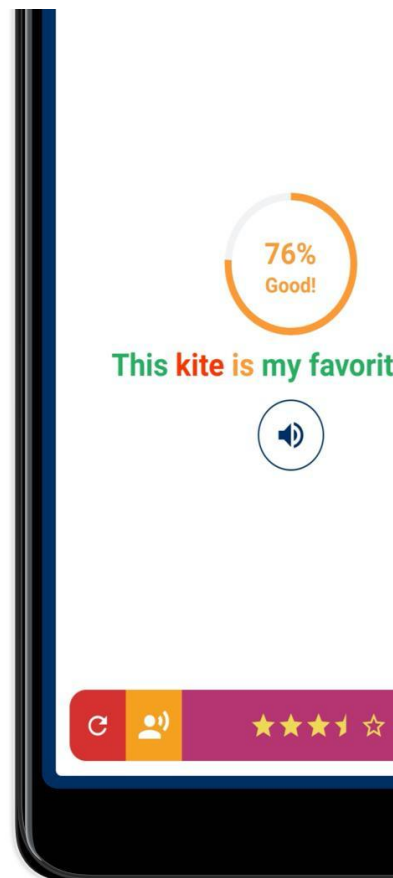
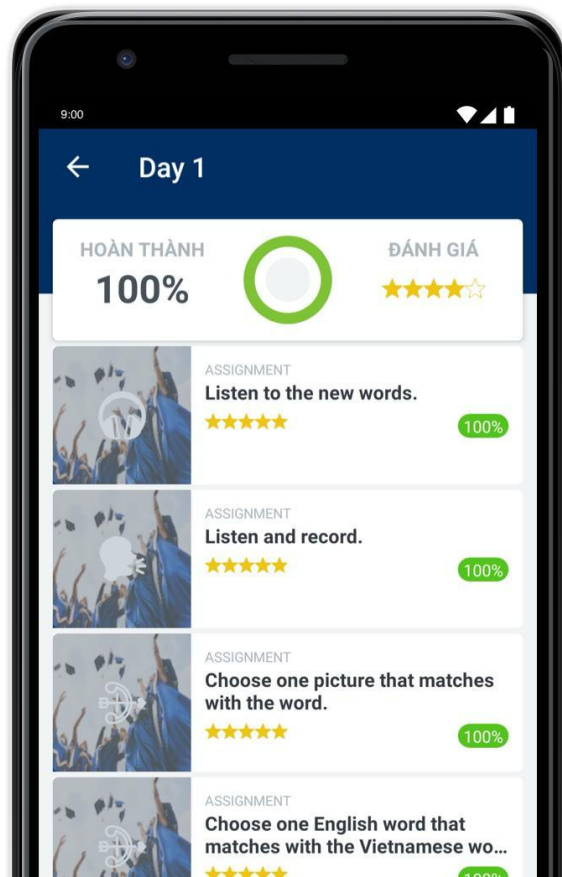


Node.js với Mobile (React Native)

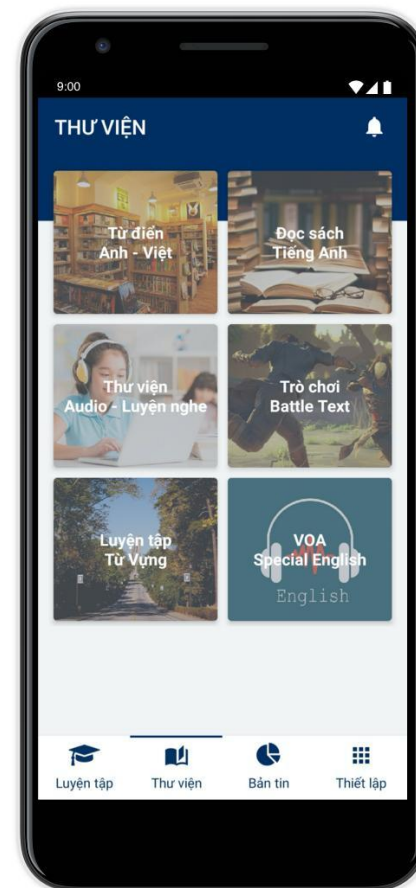


Khoá học
Tiếng Anh

Bài tập phong phú,
đa dạng



Chấm điểm ph
bằng A.I



Mở rộng vốn từ vựng
với Thư viện

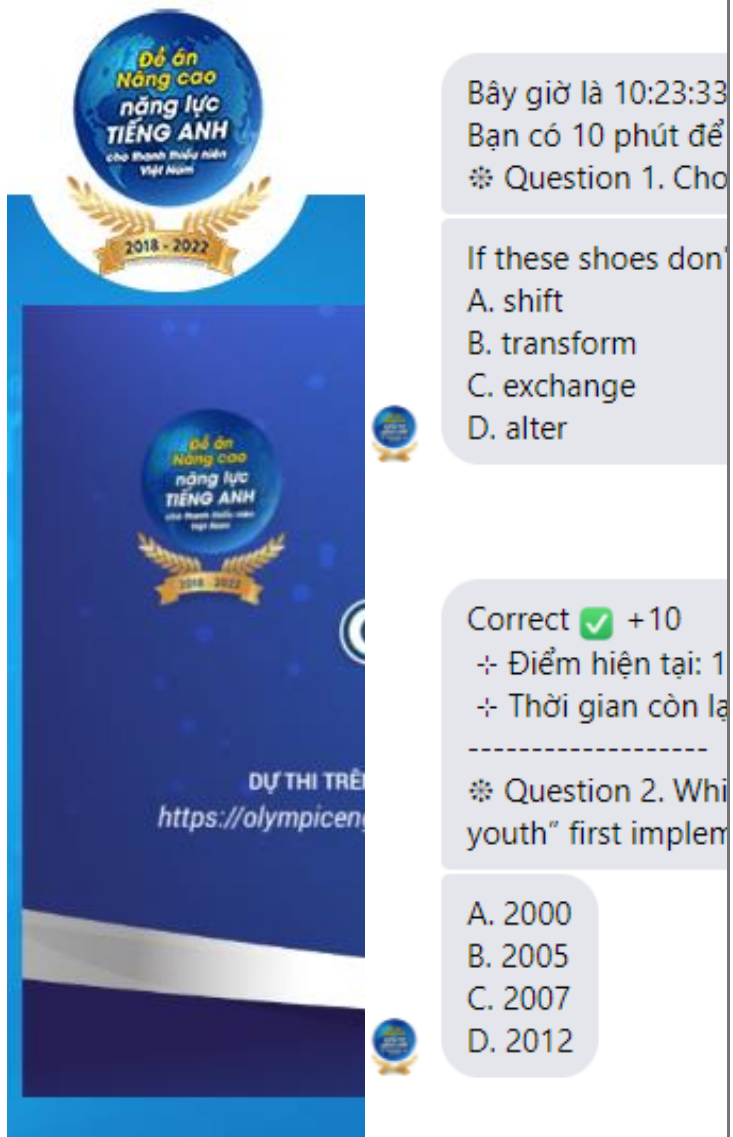
Node.js với W



PI

```
1  const { test, expect } = require('@playwright/test');
2
3  test.describe('Basic tests', () => {
4    test('Basic test 1', async ({ page }) => {
5      await page.goto('https://aptech-danang.edu.vn/');
6      await expect(page).toHaveTitle(/Softech1 Aptech2/);
7    });
8
9    test('Basic test 2', async ({ page }) => {
10     await page.goto('https://aptech-danang.edu.vn');
11     const title = page.locator('title').first();
12     await expect(title).toContainText('APTECH');
13   });
14 });
15
```


Node.js với Chatbot



Send API Basics

All messages are sent by submitting a **POST** request to the **Send API** with your page access token appended to the URL query string:

```
https://graph.facebook.com/v14.0/me/messages?access_token=<PAGE_ACCESS_TOKEN>
```

The body of the HTTP request is sent in JSON format and requires three properties:

- **messaging_type**: Identifies the purpose of the message send.
- **recipient**: Identifies the **intended recipient** of the message.
- **message**: Defines the message to be sent.

Here's a simple example of the body of a request to send a text message:

```
"messaging_type": "<MESSAGING_TYPE>",  
"recipient": {  
  "id": "<PSID>"  
},  
"message": {  
  "text": "hello, world!"  
}
```

Node.js với CMS

Select a field for your content type

 **Text** Small or long text like title or description

 **Rich Text** A rich text editor with formatting options

 **Number** Numbers (integer, float, decimal)

 **Date** A date picker with hours, minutes and seconds


 **Boolean** Yes or no, 1 or 0, true or false

 **Relation** Refers to a Content Type

 **Email** Email field with validations format

 **Password** Password field with encryption

 **Enumeration** List of values, then pick one

 **Media** Files like images, videos, etc

 **JSON** Data in JSON format

 **Component** Group of fields that you can repeat or reuse

 **Dynamic Zone** Dynamically pick component when editing ...



Node.js với Cloud



Node.js với A.I / Machine Learning



Cảm ơn

