FROM ZERO TO LIVE

How to Build Your Professional Website in a Weekend Without a Developer

— A Complete Step-by-Step Guide —

By Obed Favour Chukwuemeka

Growth Marketer & AI Automation Expert

www.obedfavour.com

Welcome

A few months ago I had no website. I had results — 1,900% YouTube growth, 3.5M+ organic views, a $1.4M pre-seed fundraise supported through community marketing — but nowhere to send people that showed all of it in one place.

So I built one. From scratch. With zero developer experience.

I used tools called Hugo and GitHub that I had never heard of before. I typed commands into PowerShell that looked like a foreign language. I hit errors that made me question everything. And I fixed every single one of them.

This guide is the exact process I followed — every command, every setting, every fix — written so simply that even someone who has never touched a terminal in their life can follow it from start to finish.

By the end of this guide you will have:

- A live professional website on your own custom domain
- A working blog that can rank on Google
- A contact form that actually sends emails
- A booking button connected to your calendar
- Google Analytics tracking every visitor
- Google Search Console indexing your pages

Total cost: Zero. Total time: One weekend.

Let's build.

What's Inside

Q

Quick Debug Reference

Every error you might hit and exactly how to fix it

CHAPTER 1

Before You Start

Everything you need before typing a single command

What You Will Build

A static website — fast, free to host, professional-looking, and fully yours. Here is the tech stack you will use:

Tool	What It Does	Cost
Hugo	Builds your website files	Free
Blowfish Theme	Makes it look beautiful	Free
GitHub	Stores your website code	Free
GitHub Pages	Hosts your website online	Free
Namecheap	Your custom domain (obedfavour.com)	~$12/year
Formspree	Contact form that sends you emails	Free
Calendly	Booking link for calls	Free
Google Analytics	Tracks your website visitors	Free

What You Need

- A Windows computer (this guide uses Windows — Mac instructions in the tips)
- A GitHub account — sign up free at github.com
- A Namecheap account — sign up free at namecheap.com
- About 6-8 hours across a weekend
- A professional photo of yourself
- Your bio, headline, and social media links ready

Before you start — do these 3 things

1. Create a GitHub account at github.com and note your username.
2. Create a repository named 'yournameweb' on GitHub (click New repository).
3. Buy your domain on namecheap.com — search for your name and buy the .com version.

CHAPTER 2

Setting Up Your Tools

Installing everything you need — takes about 30 minutes

Step 1 — Open PowerShell 7

PowerShell is a tool on your Windows computer that lets you type commands. Think of it like giving your computer instructions by typing instead of clicking. It looks scary at first — it is not. Every command in this guide is exactly what to type.

- Press the Windows key on your keyboard
- Type 'PowerShell 7' in the search bar
- Click 'Windows PowerShell 7' to open it
- You will see a blue or dark window with a blinking cursor

■ PowerShell 7 is open and ready. You will see PS C:\Users\YourName>

Step 2 — Install Scoop (Package Manager)

Scoop is a tool that helps you install other tools easily. Think of it like an app store for your command line. Type these two commands exactly:

COMMAND — Run this first:

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

When it asks you a question — type Y and press Enter.

COMMAND — Then run this:

```
irm get.scoop.sh | iex
```

This will take 1-2 minutes. You will see text scrolling. That is normal. Wait until you see the cursor again.

■ Scoop installed successfully. You will see: Scoop was installed successfully!

Step 3 — Install Hugo

Hugo is the tool that builds your website. Run this command:

COMMAND:

```
scoop install hugo-extended
```

Now verify Hugo installed correctly:

COMMAND — Check Hugo version:

```
hugo version
```

■ Hugo installed correctly. You will see: hugo v0.161.1-extended or similar.

■■ *If you see an error on Scoop install — try running PowerShell as Administrator. Right-click PowerShell and select 'Run as Administrator'.*

CHAPTER 3

Creating Your Website

Building the site structure and connecting to GitHub

Step 1 — Navigate to Your Documents Folder

Before creating anything, you need to go to the right folder. Type this command:

COMMAND:

```
cd Documents
```

■■ *If you get an error saying the path does not exist, type 'pwd' and press Enter. This shows you where you are. Then type 'cd Documents' again.*

Step 2 — Create Your Website

This command creates all the files and folders your website needs. Replace 'yournameweb' with your actual name (no spaces):

COMMAND — Replace yournameweb with your name:

```
hugo new site yournameweb
cd yournameweb
```

■ Website folder created! Hugo created your site in a folder called yournameweb.

Step 3 — Connect to GitHub

Now connect your website to GitHub so it can be hosted online. Replace 'yourusername' and 'yourrepository' with your actual GitHub details:

COMMAND:

```
git init
git remote add origin https://github.com/yourusername/yourrepository.git
```

■■ *Your GitHub repository URL is found on your GitHub repository page. It looks like: https://github.com/yourusername/yourrepository*

CHAPTER 4

Installing Your Theme

Making your website look beautiful with Blowfish

Step 1 — Install the Blowfish Theme

Blowfish is a free, professional Hugo theme with dark mode, social links, and a beautiful profile page. Run this command:

COMMAND:

```
git submodule add -b main https://github.com/nunocoracao/blowfish.git themes/blowfish
cp themes/blowfish/config/_default/* config/_default/
```

■ Blowfish theme installed! The second command copies the default settings files.

Step 2 — Configure hugo.toml

hugo.toml is the main settings file for your website. Open it in Notepad:

COMMAND:

```
notepad hugo.toml
```

Delete everything in the file and paste this exactly:

PASTE THIS INTO hugo.toml:

```
baseURL = 'https://www.yourdomainname.com/'
locale = 'en'
title = 'Your Name'
theme = 'blowfish'
googleAnalytics = "G-XXXXXXXXXX"
[markup]
[markup.goldmark]
[markup.goldmark.renderer]
unsafe = true
```

■■ *Replace 'yourdomainname.com' with your actual domain. Replace 'Your Name' with your actual name. Leave googleAnalytics blank for now — you will add it in Chapter 9.*

Step 3 — Configure params.toml

COMMAND:

```
notepad config\_default\params.toml
```

Find and change these settings (they are already in the file — just update the values):

FIND AND UPDATE THESE LINES:

```
colorScheme = "slate"
defaultAppearance = "dark"
[header]
layout = "fixed"
customCSS = ["css/custom.css"]
```

CHAPTER 5

Making It Look Like You

Adding your photo, bio, and social links

Step 1 — Add Your Profile Photo

First, create the assets folder where your photo will live:

COMMAND:

```
mkdir assets
```

Now copy your professional photo from your Downloads folder into the assets folder. Replace 'YourPhoto.jpg' with your actual photo filename:

COMMAND — Replace YourPhoto.jpg with your actual filename:

```
copy "$env:USERPROFILE\Downloads\YourPhoto.jpg" "assets\avatar.jpg"
```

■■ *Your photo MUST be saved as avatar.jpg in the assets folder — not assets/img/ or anywhere else. This is the most common mistake people make.*

Step 2 — Set Up Your Author Profile

COMMAND:

```
notepad config\_default\languages.en.toml
```

Replace everything in the file with this — filling in your own details:

PASTE THIS — FILL IN YOUR OWN DETAILS:

```
disabled = false
locale = "en"
title = "Your Name"

[params]
description = "Your Profession & Expertise"

[params.author]
name = "Your Full Name"
```

```
image = "avatar.jpg"
headline = "Your powerful headline here"
bio = "Your 1-2 sentence bio here"
links = [
{ linkedin = "https://linkedin.com/in/yourprofile" },
{ instagram = "https://instagram.com/yourhandle" },
{ x-twitter = "https://twitter.com/yourhandle" },
{ github = "https://github.com/yourusername" },
]
```

■■ *The bio line must have 2 spaces before it to match the other lines. If your profile photo or bio disappears after pushing — come back here and re-enter this file. It can reset.*

Step 3 — Preview Your Website

Before pushing anything live, preview it locally on your computer:

COMMAND:

```
hugo server
```

Open your browser and go to: http://localhost:1313

You should see your website with your photo, name, bio, and social links. Press Ctrl+C in PowerShell when you are done previewing.

■ Your website is showing locally. If you see your photo and bio — you are ready to push it live.

CHAPTER 6

Connecting Your Domain

From GitHub Pages to your own custom .com address

Step 1 — Push Your Website to GitHub

First push — this uploads everything to GitHub for the first time:

COMMANDS — Run these in order:

```
git branch -M main
git pull origin main --allow-unrelated-histories
git add .
git commit -m "initial hugo site"
git push -u origin main
```

■■ *If a black editor called Vim opens — do not panic. Type exactly :wq and press Enter. This saves and closes Vim.*

■■ *If you get 'Updates rejected' — run the git pull command again before git push.*

■ Files pushed to GitHub successfully.

Step 2 — Enable GitHub Pages

Now tell GitHub to host your website:

- ■ Go to your repository on github.com
- ■ Click Settings (top right of the repository)
- ■ Click Pages in the left sidebar
- ■ Under Source — select GitHub Actions
- ■ Click Save

■ Your site is now being deployed. Wait 2-3 minutes and it will be live at yourusername.github.io/yourrepository

Step 3 — Create Your CNAME File

This file tells GitHub which domain to use. Do NOT copy and paste this — type it manually:

COMMAND — Open Notepad:

```
notepad static\CNAME
```

In Notepad — manually type (do not paste):

TYPE THIS MANUALLY:

```
www.yourdomainname.com
```

Save and close Notepad. Then verify it looks correct:

COMMAND — Verify:

```
notepad static\CNAME
```

■■ *IMPORTANT: The CNAME file must contain PLAIN TEXT only. If it shows www.yoursite.com with brackets — the file has a hyperlink not plain text. Delete and retype manually.*

Step 4 — Set Up DNS on Namecheap

Go to namecheap.com, log in, click Domain List, click Manage next to your domain, then click Advanced DNS. Add these records exactly:

Type	Host	Value
A Record	@	185.199.108.153
A Record	@	185.199.109.153
A Record	@	185.199.110.153
A Record	@	185.199.111.153
CNAME Record	www	yourusername.github.io
URL Redirect	@	https://www.yourdomainname.com (Permanent 301)

Step 5 — Connect Domain on GitHub

■ Go to your repository on GitHub

- Click Settings → Pages
- Under Custom domain — type: www.yourdomainname.com
- Click Save and wait for DNS check to complete (up to 30 minutes)
- Once DNS check passes — tick Enforce HTTPS

- Your website is now live at www.yourdomainname.com with HTTPS enforced.

CHAPTER 7

Creating Your Pages

Building your About, Portfolio, Services, and Blog pages

Step 1 — Set Up Your Navigation Menu

COMMAND:

```
notepad config\_default\menus.en.toml
```

Delete everything and paste this:

PASTE THIS:

```
[[main]]
name = "Home"
url = "/"
weight = 10
[[main]]
name = "About"
pageRef = "about"
weight = 20
[[main]]
name = "Portfolio"
pageRef = "portfolio"
weight = 25
[[main]]
name = "Blog"
pageRef = "posts"
weight = 30
[[main]]
name = "Services"
pageRef = "services"
weight = 40
```

Step 2 — Create Your Page Files

COMMANDS — Run all of these:

```
hugo new content/about.md
hugo new content/portfolio.md
hugo new content/services.md
hugo new content/posts/_index.md
```

■ Page files created in the content folder. Now open each one in Notepad and add your content.

Step 3 — Edit Your About Page

COMMAND:

```
notepad content\about.md
```

Your about page uses Markdown — a simple way to format text. Here is a template:

ABOUT PAGE TEMPLATE:

```
+++
title = "About"
draft = false
+++

## Who I Am
Write your story here. Who are you and what do you do?

## What I Do
- Service 1
- Service 2
- Service 3

## Let's Work Together
[Book a call →](https://calendly.com/yourlink)
```

Step 4 — Push Your Pages Live

COMMANDS — Do this after every set of changes:

```
git add .
```

```
git commit -m "add pages"
git push
```

Wait 60-90 seconds and check your live website.

CHAPTER 8

Writing Your First Blog Post

Publishing content that Google will find and rank

Step 1 — Create the Blog Post File

COMMAND — Replace the filename with your post title:

```
hugo new content/posts/my-first-blog-post.md
```

■■ *Use hyphens between words in your filename (not spaces). This becomes your URL — keep it short and keyword-rich.*

Step 2 — Write Your Post

COMMAND:

```
notepad content\posts\my-first-blog-post.md
```

Your post file has a front matter section (between the +++ marks) and then your content below:

BLOG POST TEMPLATE:

```
+++
title = "Your Blog Post Title Here"
date = '2026-01-01T10:00:00+01:00'
draft = false
description = "A 1-2 sentence description of your post for Google"
tags = ["marketing", "growth", "your topic"]
+++

Your introduction paragraph goes here.

## Your First Section Heading

Write your content here.

## Your Second Section Heading

More content here.
```

■■ *IMPORTANT: Make sure draft = false (not draft = true). If draft = true, your post will not show up on your website.*

■■ *Make sure the date is not in the future. Hugo hides future-dated posts. Set it to today's date.*

Step 3 — Add Images to Your Blog Post

Download a free image from unsplash.com. Save it to your Downloads folder with a simple name like hero.jpg. Then copy it to your site:

COMMANDS:

```
mkdir static\blog
copy "$env:USERPROFILE\Downloads\hero.jpg" "static\blog\hero.jpg"
```

Then add the image to your blog post using this shortcode:

ADD THIS TO YOUR BLOG POST:

```
{{< figure src="/blog/hero.jpg" alt="Description of your image" >}}
```

CHAPTER 9

Getting Found on Google

Analytics, Search Console, sitemap and indexing your pages

Step 1 — Set Up Google Analytics

- Go to analytics.google.com and sign in with your Google account
- Click Start measuring
- Create an account name and property name (your website name)
- Choose Web as your platform
- Enter your website URL and click Create stream
- Copy your Measurement ID — it looks like G-XXXXXXXXXX

Now add it to your hugo.toml file:

COMMAND:

```
notepad hugo.toml
```

Find the googleAnalytics line and add your ID:

UPDATE THIS LINE:

```
googleAnalytics = "G-XXXXXXXXXX"
```

■■ *The googleAnalytics line must be in hugo.toml — NOT in params.toml. This is the most common Analytics setup mistake.*

Step 2 — Create robots.txt

COMMAND:

```
notepad static\robots.txt
```

Type this exactly:

TYPE THIS:

```
User-agent: *
Allow: /
```

```
Sitemap: https://www.yourdomainname.com/sitemap.xml
```

Step 3 — Push and Set Up Search Console

COMMANDS:

```
git add .
git commit -m "add analytics and robots.txt"
git push
```

- ■ Go to search.google.com/search-console
- ■ Click Add Property → URL Prefix → enter https://www.yourdomainname.com
- ■ Verify using Google Analytics (instant if GA is set up)
- ■ Click Sitemaps in the left menu → type sitemap.xml → Submit
- ■ Click URL Inspection → paste your homepage URL → Request Indexing
- ■ Repeat for each page: /about, /portfolio, /services, /posts

■ Google Search Console is set up. Your pages are queued for indexing. Allow 24-72 hours.

CHAPTER 10

The Finishing Touches

Calendly, contact form, favicon, and mobile responsive CSS

Step 1 — Add a Booking Button (Calendly)

Calendly lets people book calls with you directly from your website. Sign up free at calendly.com. Once you have your booking link, add this button to any page:

ADD THIS TO ANY .md PAGE FILE:

```
<div style="text-align:center;margin:2rem 0;">
<a href="https://calendly.com/yourusername/30min"
style="display:inline-block;background:#2E75B6;color:#fff;
padding:14px 32px;border-radius:6px;font-size:15px;
font-weight:500;text-decoration:none;">Book a Free Call</a>
</div>
```

Step 2 — Add a Contact Form (Formspree)

Formspree sends you an email every time someone fills in your contact form. Sign up free at formspree.io. Get your form endpoint (looks like https://formspree.io/f/xxxxxxxx) then add this to your services page:

ADD TO YOUR SERVICES PAGE:

```
<form action="https://formspree.io/f/XXXXXXXX" method="POST"
style="display:flex;flex-direction:column;gap:1rem;max-width:600px;">
<input type="text" name="name" placeholder="Your Name" required
style="padding:10px;border-radius:6px;border:1px solid #444;">
<input type="email" name="email" placeholder="Your Email" required
style="padding:10px;border-radius:6px;border:1px solid #444;">
<textarea name="message" rows="5" required
style="padding:10px;border-radius:6px;border:1px solid #444;"></textarea>
<button type="submit" style="padding:12px 28px;background:#2E75B6;
color:#fff;border:none;border-radius:6px;cursor:pointer;">Send</button>
```

```
</form>
```

Step 3 — Add Custom CSS for Full Width and Mobile

COMMAND:

```
mkdir assets\css
notepad assets\css\custom.css
```

Paste this CSS:

PASTE THIS:

```
article { max-width: 100% !important; }
.prose { max-width: 100% !important; }
.max-w-prose { max-width: 100% !important; }
.content { max-width: 100% !important; }
@media (max-width: 640px) {
.prose div[style*='grid-template-columns:repeat(4'] {
grid-template-columns: repeat(2, 1fr) !important;
}
input, textarea, select {
width: 100% !important;
box-sizing: border-box !important;
}
}
```

Step 4 — Final Push

COMMANDS:

```
git add .
git commit -m "add finishing touches"
git push
```

■ Your website is fully live, connected to Google, and ready to receive visitors and enquiries.

Glossary

Plain-English explanations of every technical term in this guide.

Git

A system that saves every version of your website code. Think of it like a super-powered undo button that remembers every change you have ever made.

GitHub

A website where your website code is stored online. Like Google Drive but specifically for code. It also hosts your website for free through GitHub Pages.

GitHub Pages

GitHub's free website hosting service. When you push your code to GitHub, GitHub Pages automatically builds and hosts your website online.

GitHub Actions

An automation tool inside GitHub. Every time you push new code, GitHub Actions automatically rebuilds and redeploys your website. You set it up once and it runs forever.

Hugo

A static site generator — a tool that takes your content files and turns them into a complete website. It builds your site in milliseconds and produces files that load extremely fast.

Blowfish

A free Hugo theme — a pre-designed visual template that makes your website look beautiful without you having to design anything from scratch.

Terminal / PowerShell

A text-based window where you give your computer instructions by typing commands instead of clicking. Looks scary but is just another way to control your computer.

Command

An instruction you type into PowerShell to make your computer do something. Like telling your computer what to do in its own language.

Repository (Repo)

A folder on GitHub that contains all your website files. Think of it as your website's home on GitHub.

Push

Sending your local changes to GitHub. When you 'push' — your latest version of the website goes to GitHub and gets deployed live.

Commit

Saving a snapshot of your changes with a message describing what you changed. Like saving a document with a note about what you edited.

Branch

A version of your repository. 'Main' is the primary branch — the live version of your site. You will always work on main in this guide.

DNS

Domain Name System — the internet's address book. DNS records tell the internet where to find your website when someone types your domain name.

Domain

Your website address — like www.yourname.com. You register this on Namecheap and connect it to your GitHub Pages website.

CNAME

A DNS record type that points your www domain to your GitHub Pages address. Also a file in your static folder that tells GitHub Pages which domain to use.

HTTPS

The secure version of HTTP. The padlock icon you see in browser address bars. Enforcing HTTPS means your site is secure and Google will rank it better.

SSL Certificate

A digital certificate that enables HTTPS. GitHub Pages creates this for you automatically once your domain is connected.

Static Site

A website made of simple HTML files — no database, no server-side code. Fast, secure, and free to host. Everything in this guide builds a static site.

Markdown

A simple way to format text using symbols. A # makes a heading. **bold** makes text bold. A - makes a bullet point. Hugo reads Markdown files and turns them into website pages.

Front Matter

The section at the top of every content file between the +++ marks. It contains settings like the page title, date, and whether the page is a draft.

Draft

A setting in your front matter. draft = true means the page is hidden. draft = false means it is live. Always check this if a page is not showing up.

Hugo Theme

A collection of design files that control how your website looks. Blowfish is the theme used in this guide.

Submodule

A way of including another GitHub repository inside yours. Blowfish is added as a submodule — it links to the Blowfish repository so you always get updates.

baseURL

The main web address of your site. Set in hugo.toml. Must match your custom domain exactly — including https:// and the trailing slash.

Favicon

The small icon that appears in your browser tab next to your website title. The guide uses favicon.io to generate one with your initials.

Google Analytics

Google's free tool for tracking website visitors — who visits, where they come from, which pages they read, and how long they stay.

Google Search Console

Google's free tool for managing your website's presence in search results. Use it to submit your sitemap and request indexing of your pages.

Sitemap

A file (sitemap.xml) that lists every page on your website. You submit it to Google Search Console so Google knows what pages to index.

robots.txt

A file that tells search engine crawlers which pages they are allowed to index. Your robots.txt tells Google it can crawl everything.

Indexing

The process of Google adding your page to its search database so it can appear in search results. A page must be indexed to appear on Google.

Canonical URL

The official, preferred version of a page URL. If your site has multiple versions (http, https, www, non-www) — the canonical URL is the one Google should index.

CSS

Cascading Style Sheets — the code that controls how your website looks. Colours, fonts, layout, and spacing are all controlled by CSS.

Formspree

A free service that processes your contact form submissions and sends them to your email. No server needed.

Calendly

A free scheduling tool that lets people book calls with you directly. You connect it to your calendar and share a link.

Scoop

A free package manager for Windows — a tool that helps you install other tools easily via PowerShell commands.

Vim

A text editor that can open unexpectedly when running Git commands. If it opens — type :wq and press Enter to close it safely.

Quick Debug Reference

Every error you might hit and the exact fix. Bookmark this page.

ERROR: cd : Cannot find path ... because it does not exist

FIX: You are already in the folder. Run 'pwd' to see where you are. If it shows obedfavourweb — skip the cd command.

ERROR: src refspec main does not match any

FIX: Run: git branch -M main — then try git push again.

ERROR: Updates rejected — remote contains work not in local

FIX: Run: git pull origin main --allow-unrelated-histories — then git push again.

ERROR: Vim opened and you are stuck

FIX: Type exactly :wq and press Enter. This saves and closes Vim.

ERROR: Profile photo not showing

FIX: Make sure your photo is in the assets/ folder (not assets/img/). The file must be named avatar.jpg. Check that image = 'avatar.jpg' is set in languages.en.toml.

ERROR: Profile info disappeared (name, bio, headline)

FIX: The languages.en.toml file reset. Open it with: notepad config_default\languages.en.toml — and re-enter your details. Make sure bio has 2 spaces before it.

ERROR: Blog post not showing on the website

FIX: Check two things: 1) draft = false in the post file. 2) The date is not set in the future. Hugo hides future-dated posts.

ERROR: HTML not rendering in markdown pages

FIX: Make sure hugo.toml has unsafe = true under [markup.goldmark.renderer]. This is required for raw HTML in .md files.

ERROR: CNAME file saving as hyperlink

FIX: Do not copy and paste the domain from anywhere. Open Notepad manually and TYPE the domain character by character: w-w-w-.-y-o-u-r-d-o-m-a-i-n-.-c-o-m

ERROR: Google Analytics not tracking

FIX: Make sure googleAnalytics is in hugo.toml — NOT in params.toml. Only hugo.toml works for Blowfish.

ERROR: Blog images not showing

FIX: Use the Hugo figure shortcode: {{< figure src='/blog/image.jpg' alt='description' >}} — Images must be in the static/blog/ folder.

ERROR: Changes not showing on live site

FIX: Run: git add . then git commit -m 'your message' then git push. Wait 60-90 seconds. The site rebuilds automatically after every push.

ERROR: Domain not loading

FIX: Check Namecheap DNS records match exactly. Check GitHub Pages has your custom domain set and HTTPS is enforced. Check static/CNAME contains plain text only.

ERROR: Enforce HTTPS greyed out on GitHub

FIX: The SSL certificate has not been issued yet. Wait 15-30 minutes after setting your custom domain. The option will become clickable once the certificate is ready.

ERROR: Digital Products page going to /posts/ instead of /products/

FIX: Open content/products.md and make sure: draft = false and slug = 'products' are set in the front matter.

You Built It.

You now have a professional website, a working blog, Google Analytics, Search Console, a contact form, and a booking link — all for free.

What's next:

- Write your first blog post and request indexing in Search Console
- Share your website link on LinkedIn and tell people what you built
- Add your Calendly link everywhere and start booking calls
- Check Google Analytics every Monday — watch the visitors grow

---

About the Author

Obed Favour Chukwuemeka is a Growth Marketer and AI Automation Expert based in Lagos, Nigeria. He helps founders, startups, and growing brands build marketing systems that scale. He built obedfavour.com from scratch using exactly the steps in this guide.

www.obedfavour.com · linkedin.com/in/obed-favour-speaks