Hello! My name is

Vadim Cherepenichev

and I'm a **Senior WordPress Developer**

with over **8 years of experience**

TOP CLIENTS

TopTal

A platform connecting top developers with top clients. TopTal has a highly selective 5-steps screening process.

Trust Payments

An international payment processor.

Joslex Studios

A US-based WordPress development agency that ranked among Google's top 10 search results in 2021.

Vet Clinics Network

The largest vet clinics network in a Western European country. The company name is under NDA.

and 17 other agencies and clients across the US and Europe!

SKILLS

WordPress

Fundamentals

WP_Query, $wpdb, REST, Cron, Hooks, Options, Shortcodes, Widgets, Custom Post Types / Taxonomies / User Roles

Advanced Core Concepts

Multisite, Multilingual, RTL, Transients, Permalinks API, WP CLI

Custom Plugins and Themes

from scratch / WooCommerce based / ACF based

Gutenberg, BeaverBuilder, BB Themer, Elementor, WPBackery, Divi, Astra

and custom blocks / modules development

WP as a Headless CMS

General

JS

TypeScript, React, Vue, Node, Anime, D3, WebWorkers, WebSockets

CSS

Sass, Less, Stylus / BEM

Databases

MySQL, MariaDB, PostgreSQL, Elasticsearch

Frontend Optimization

Localstorage, IndexedDB, Lazyload, Critical CSS, Minification, GZIP, CDN, Chrome DevTools Profiler, Firebug

Backend Optimization

Varnish, Redis, Memcached, XDebug Profiler

Architecture / Patterns

Microservices, MVC

Basic

HTML / CSS / JS

Cross-browser compatibility, pixel-perfect design, responsiveness, semantic HTML, accessibility

Version Control

Git, SVN, GitHub, GitLab

CLI, SSH, Linux

Paradigms

DRY, KISS, YAGNI, OOP

Communication & PM Tools

Jira, Asana, Monday, Trello, Slack

Design Tools

Photoshop, Figma, Illustrator, Sketch

using REST API or GraphQL

Knowledge of Popular Plugins

including their hooks

Thirdparty Services Integration

using existing plugins or PHP SDKs

Debugging / Testing

APM, Postman, PHPUnit, XDebug, Nightwatch.js, Mocha, Chai, Sinon

Build Tools

Composer, Webpack, Gulp, Grunt

DevOps

Docker, Capistrano

Web Servers

NGINX, Apache, Express.js

WP PROJECTS

WP Theme / Plugin Boilerplate

Exactly what you're looking for if you want to see my code!

See this GitHub repository.

Problem

The 2 agencies I worked at had a problem with codebase standardization.

Reasons:

- No well-defined internal standards or guidelines.
- Many small to medium-sized projects.
- Developers writing code and organizing projects in their own way.
- Frequent developer switching between projects due to time constraints of the original code authors.

Each new project required developers to spend time familiarizing themselves with the codebase, which was often inconsistent and poorly documented. This made it difficult to maintain and scale.

Solution

I created a boilerplate that maintained flexibility and did not restrict developers in their work. The purpose of the boilerplate was to prevent desynchronization between different projects in the file structure and codebase for core WordPress functionalities like shortcodes, metaboxes, post types, etc.

This boilerplate, along with new project management approaches introduced by the agencies, led to a significant improvement in codebase quality and maintainability. Developers were no longer spending time on repetitive basic tasks, and newly onboarded team members could start working on projects faster since they were already familiar with the file structure and basic codebase.

The boilerplate includes/implements:

- Comprehensive instructions in the `README.md`.
- Helper classes for common WordPress tasks:
  - Shortcodes
  - Post types
  - Posts
  - Taxonomies
  - Metaboxes
  - Assets management
  - Etc.
- A site setup validator for plugin dependencies, constants, settings, etc.
- Composer integration and PHP class autoloading.
- Standardized file structure for:
  - CSS assets
  - JS assets
  - Templates
- Tools for asset source file watching (SCSS, TypeScript), compilation, and bundling (Webpack, Gulp).
- WP All-in-One Migration preset configurations.

- Astra/Beaver Builder settings synchronization (e.g., buttons, responsive breakpoints, etc.).
- And several other features (see the GitHub repository).

WP Multisite for Multi-Department Company

Problem

The company I developed this project for had multiple departments, each with unique requirements. However, much of the business logic and design elements needed to be shared across all sites within the network.

Solution

To address these needs, I implemented a multisite network with the following components:

1. *Global Core Plugin* - a plugin applied to the entire network.
2. *Department-Specific Core Plugins* - one plugin for each department's sub-site.
3. *Global Parent Theme* - a shared theme for common design elements.
4. *Department-Specific Child Themes* - tailored themes for each sub-site.

Why Multisite?

A multisite network offered several advantages:

- Streamlined Codebase and Management. It simplified both the code structure and the settings management.
- Efficiency. While it's technically possible to achieve similar results without multisite using various CD tools, a multisite setup provided a significantly faster solution.
- No Typical Drawbacks. Issues often associated with multisite, such as a shared database, were not relevant to this project.

Why Separate Code Between Plugins and Themes?

Plugins act as the Model. Responsible for business logic and core functionalities.

Themes act as the View. Focused on the design and user interface.

This separation adhered to development best practices, ensuring clarity and maintainability for the considerable amount of code involved.

Why Split Logic Between Core and Department-Specific Components?

Simplified Maintenance. Isolating department-specific code minimizes the risk of errors. For instance:

- Code for Department A is not activated on Department B's site.
- Changes to Department C's code won't affect Departments A or B.

Flexibility. Each department can:

- Utilize its own technologies.
- Use different versions of the same libraries without conflict.

This architecture ensured a robust, scalable, and efficient solution tailored to the company's diverse needs.

Knowledge Base with Instant-Autocomplete Search for 10M+ Items

Problem

A knowledge base website with a filtering feature that includes several fields. One of the key components is a text input for searching by content.

Solution

ElasticSearch Integration

Large text data is stored in Elasticsearch to minimize the size of the SQL database (MariaDB).

SQL Database

- Structure. The SQL database stores only item titles and meta-information. Data is distributed across three tables for efficiency:

  - Primary Data Table: Contains essential information, such as titles and core details.
  - Filterable Data Table: Stores data specifically used for filtering items.
  - Metadata Table: Contains information not relevant to filtering.
- Indexes for Queries Optimization:
  - The filterable data table is equipped with multiple indexes, including composite indexes tailored for specific query types.
  - Although some composite indexes overlap, resulting in increased storage use and slower write times, this trade-off is acceptable because the database workload is predominantly read-heavy.
- Data Joins:
  - For complete query results, data from the primary and metadata tables is joined with filterable table data using.
- Partitioning:
  - Data is devided via `PARTITION` directive into smaller, more manageable parts for faster access.

Caching

- Redis: Stores the most frequently accessed item data and filter search results.
- Frontend Local Storage: Caches data on the client side.

AI

A real case study: I leveraged my AI skills to cut project development time by automating a task that was originally estimated to take 2 weeks. I got it done in just 3 hours using a bit of coding and the OpenAI API.

Advanced Prompting / OpenAI API / ChatGPT Pro Subscription / Copilot Subscription

WP WORK EXPERIENCE

Joslex Studios

Senior Full-Stack Developer

Key Responsibilities

- Implementation of a read-heavy database project (4 million filterable items)
- Implementation of multisite projects
- Optimization of existing themes and plugins
- Development of themes based on ACF and Page Builders

TopTal

Senior Full-Stack Developer

TopTal is an exclusive network of the world's top freelance software developers and IT professionals. Leading companies rely on TopTal freelancers for their most critical projects.

To join TopTal, developers must pass a rigorous 5-step screening process. I was accepted at the age of 23, which the onboarding manager described as a notable achievement.

Key Responsibilities

- Direct client communication
- Development of projects with highly stringent requirements for page load time and rendering performance
- Development of e-commerce projects with tricky business logic

Trust Payments

Senior Frontend Developer

Trust Payments is a payments provider with main office in London, UK.

I'm proud to have worked with them! The job description is under NDA.

The Largest Vet Clinics Network in a Western European Country

Senior Full-Stack Developer

The project demanded strict adherence to performance metrics, including page load speed and minimizing CLS (Cumulative Layout Shift) optimization.

Key Responsibilities

- Optimization of TTFB (Time to First Byte)
- Optimization of LCP (Largest Contentful Paint)
- Optimization of CLS (Cumulative Layou Shift)
- Profiling and optimization of Client-side code (JS)
- Profiling and optimization of Server-side code (PHP)
- Implementation of new features

Digital Pulp

Senior Full-Stack Developer

Key Responsibilities

- Development of a custom ACF-based theme
- Integration of Twig template engine
- Implementation of client-side localstorage-based cache
- Adjusting routing via the WordPress Permalinks API

CONTACTS & LINKS

vadim.cvy@gmail.com github.com/vadim-cvy