

الحمد لله
الرحمن الرحيم



دانشگاه جامع عملی کاربردی

موسسه آموزش عالی آزاد انفورماتیک ایران

پایان نامه جهت دریافت درجه کارشناسی (B.S)
در رشته مهندسی فناوری گرایش برنامه‌سازی وب

عنوان:

بررسی فرآیند بومی‌سازی قالب‌ها و افزونه‌های وردپرس برای بازار ایران

استاد راهنما:

مهندس نوید همراهی

نگارش:

علی ستارزاده

بهار ۱۴۰۴

سپاس‌گذاری

مهندس نوید همراهی، دوستان و همکارانی که در نگارش این پروژه راهنمایی کردند.

تقدیم به

پدر و مادر عزیزم که همواره حامی من بوده‌اند.

فهرست مطالب

چکیده.....	۱
فصل اول: کلیات تحقیق.....	۲
(۱-۱) مقدمه.....	۳
(۲-۱) بیان مسأله.....	۳
(۳-۱) اهمیت و ضرورت تحقیق.....	۵
(۴-۱) اهداف تحقیق.....	۶
(۱-۴-۱) هدف اصلی.....	۶
(۲-۴-۱) اهداف فرعی.....	۶
(۵-۱) سوالات تحقیق.....	۷
(۱-۵-۱) سؤال اصلی.....	۷
(۲-۵-۱) سوالات فرعی.....	۷
(۶-۱) روش تحقیق.....	۸
(۷-۱) تعریف واژگان کلیدی.....	۹
فصل ۲: مروری بر ادبیات تحقیق و پیشینه تحقیق.....	۱۱
(۱-۲) معرفی CMS وردپرس.....	۱۲
(۱-۱-۲) تاریخچه و آمار استفاده از وردپرس.....	۱۲
(۲-۱-۲) مزایا و معایب استفاده از وردپرس.....	۱۳
(۲-۲) ساختار قالب‌ها و پلاگین‌ها در وردپرس.....	۱۴
(۱-۲-۲) ساختار فایل‌های قالب وردپرس.....	۱۵
(۲-۲-۲) ساختار فایل‌های افزونه وردپرس.....	۱۶
(۳-۲) مفهوم بومی‌سازی (Localization) در نرم‌افزار و وب.....	۱۷
(۱-۳-۲) تعریف بومی‌سازی و تمایز آن با ترجمه (Translation) و جهانی‌سازی (Globalization).....	۱۷
(۲-۳-۲) اهمیت بومی‌سازی در نرم‌افزار و وب.....	۱۸
(۴-۲) مروری بر مارکت‌پلیس‌های ایرانی (مثل ژاکت، راست‌چین).....	۱۹
(۱-۴-۲) معرفی مارکت‌پلیس‌های ایرانی فعال در حوزه وردپرس.....	۲۰

۲-۴-۲	بررسی ویژگی‌ها و خدمات آن‌ها.....	۲۱
۵-۲	مقالات یا پروژه‌های مشابه در این حوزه.....	۲۲
۱-۵-۲	بررسی مقالات و پایان‌نامه‌های مرتبط.....	۲۲
۲-۵-۲	تحلیل یافته‌های آن‌ها و شکاف تحقیقاتی.....	۲۳
فصل ۳:	روش اجرای تحقیق و تجزیه و تحلیل داده‌ها (یافته‌ها).....	۲۵
۱-۳	انتخاب یک قالب از مارکت بین‌المللی (مطالعه موردی: ThemeForest).....	۲۶
۱-۱-۳	معرفی ThemeForest و معیارهای انتخاب قالب.....	۲۶
۲-۱-۳	ملاحظات مهم در انتخاب قالب با هدف بومی‌سازی.....	۲۷
۲-۳	راه‌اندازی محیط توسعه‌ی محلی و نصب وردپرس.....	۲۹
۱-۲-۳	اهمیت محیط توسعه‌ی محلی.....	۲۹
۲-۲-۳	نصب و راه‌اندازی Laragon.....	۳۰
۴-۲-۳	نصب وردپرس در محیط محلی (برای شروع از صفر).....	۳۲
۳-۳	بررسی ساختار فایل‌های قالب، ترجمه متون، فونت و RTL.....	۳۲
۱-۳-۳	ساختار فایل‌های قالب انتخابی (تراولر).....	۳۲
۲-۳-۳	دانش پایه HTML و CSS مورد نیاز برای بومی‌سازی.....	۳۳
۳-۳-۳	رویکرد راست‌چین‌سازی قالب استاتیک.....	۳۳
۴-۳	ابزارهای مورد استفاده (Poedit، Loco Translate، ویرایشگر کد).....	۳۵
۱-۴-۳	معرفی و کاربرد ابزارهای ترجمه (Poedit، Loco Translate).....	۳۵
۲-۴-۳	ویرایشگرهای کد (مثل VS Code) و اهمیت آن‌ها.....	۳۶
۳-۴-۳	کار با Composer و افزودن DD به پروژه برای دیباگ.....	۳۷
۵-۳	مستندسازی فرآیند فارسی‌سازی قالب (مطالعه موردی: قالب تراولر).....	۳۸
۱-۵-۳	نصب قالب تراولر و افزونه‌های مورد نیاز.....	۳۸
۲-۵-۳	رفع تحریم لایسنس بین‌الملل.....	۳۸
۳-۵-۳	بررسی تقویم و جایگزینی فایل‌های daterangepicker و moment.....	۴۰
۴-۵-۳	شمسی‌سازی کامل تقویم (js, css).....	۴۱
۵-۵-۳	افزودن jdatetime به قالب و سفارشی‌سازی و توسعه‌ی آن.....	۴۴

۴۵	۳-۵-۶) افزودن قابلیت تبدیل تاریخ در منطق قالب به وسیله‌ی متود اختصاصی (PHP)
۴۸	۳-۵-۷) ترجمه محتوای استاتیک و پست تایپ‌های قالب
۴۸	۳-۵-۸) بررسی فایل‌های ترجمه و تکنیک‌های مرتبط
۴۹	۳-۵-۹) بررسی فایل CSS راست‌چین و افزودن فونت
۵۰	۳-۵-۱۰) سناریو تعریف فونت در ریداکس
۵۰	۳-۶) حل مشکلات مربوط به راست‌چین‌سازی (CSS و فایل‌های قالب)
۵۱	۳-۷) تست نهایی و آماده‌سازی برای انتشار
۵۱	۳-۷-۱) ایجاد پیش‌نمایش پوسته درون هاست
۵۲	۳-۸) ثبت در مارکت ایرانی (مطالعه موردی: راست‌چین)
۵۲	۳-۸-۱) ایجاد لندینگ و محتوای متناسب برای انتشار در راست‌چین
۵۴	۳-۸-۲) ارسال به راست‌چین (تمامی مراحل + لایسنس‌گذاری روی قالب)
۵۵	۳-۸-۳) کسب درآمد از فروش قالب در مارکت پلیس‌های ایرانی
۵۹	فصل ۴: نتیجه‌گیری و پیشنهادات کارهای آتی
۶۰	۴-۱) جمع‌بندی اقدامات انجام‌شده
۶۲	۴-۲) چالش‌های encountered در پروژه
۶۴	۴-۳) پیشنهادهایی برای آینده
۶۴	۳-۴-۱) خودکارسازی فرآیند بومی‌سازی
۶۵	۴-۳-۲) ساخت قالب‌ها و افزونه‌های اختصاصی بومی
۶۵	۴-۳-۳) بهبود همکاری با مارکت پلیس‌ها
۶۶	۴-۳-۴) تحقیقات آتی در حوزه بومی‌سازی وردپرس
۶۷	فهرست منابع
۶۸	Abstract

فهرست اشکال

۱۲	تصویر ۲-۱: لوگوی وردپرس
۱۶	تصویر ۲-۲: ساختار فایل‌های وردپرس
۲۰	تصویر ۲-۳: شرکت ژاکت
۲۱	تصویر ۲-۴: شرکت راست چین
۲۶	تصویر ۳-۱: سایت THEMEFOREST
۲۹	تصویر ۳-۲: قالب TRAVELER
۳۰	تصویر ۳-۳: محیط LARAGON
۳۱	تصویر ۳-۴: نصب DUPLICATOR
۳۴	تصویر ۳-۵: فایل RTL-STYLE.CSS
۳۵	تصویر ۳-۶: تصویر محیط برنامه POEDIT
۳۶	تصویر ۳-۷: افزونه‌ی وردپرس LOCO TRANSLATE
۳۹	تصویر ۳-۸: مسدود کردن API های مربوط به لایسنس
۴۱	تصویر ۳-۹: دریافت تقویم جلالی از گیت‌هاب
۴۴	تصویر ۳-۱۰: فارسی سازی CSS
۴۶	تصویر ۳-۱۱: متود اختصاصی تبدیل تاریخ
۵۳	تصویر ۳-۱۲: پیش نمایش قالب روی سایت راست چین
۵۴	تصویر ۳-۱۳: ایجاد اسکرین شات‌های با کیفیت در صفحه‌ی لندینگ
۵۶	تصویر ۳-۱۴: درآمد فروشندگان راست چین
۵۷	تصویر ۳-۱۵: فروشندگان راست چین
۵۷	تصویر ۳-۱۶: فروشندگان راست چین
۵۷	تصویر ۳-۱۷: پر فروش ترین قالب‌های راست چین
۵۸	تصویر ۳-۱۸: پر فروش ترین قالب‌های راست چین

چکیده

در عصر حاضر، وبسایت‌ها نقشی محوری در ارتباطات، تجارت و زندگی روزمره‌ی افراد ایفا می‌کنند. وردپرس (WordPress) به عنوان محبوب‌ترین سیستم مدیریت محتوا در جهان، بستری قدرتمند و انعطاف‌پذیر برای توسعه‌ی انواع وبسایت‌ها فراهم آورده است. با این حال، بسیاری از قالب‌ها و افزونه‌های وردپرس با تمرکز بر بازارهای بین‌المللی توسعه می‌یابند که این امر، چالش‌هایی را در زمینه‌ی سازگاری با نیازهای کاربران فارسی‌زبان و فرهنگ ایرانی ایجاد می‌کند. این پایان‌نامه با هدف بررسی جامع و ارائه‌ی یک مدل عملیاتی برای فرآیند بومی‌سازی قالب‌ها و افزونه‌های وردپرس جهت عرضه در مارکت‌پلیس‌های ایرانی نگاشته شده است.

روش تحقیق این پژوهش، ترکیبی از مطالعه‌ی ادبیات و یک مطالعه‌ی موردی (Case Study) عملی است. در بخش عملی، یک قالب بین‌المللی پرطرفدار با نام "تراولر" (Traveler) انتخاب شد و فرآیند کامل بومی‌سازی آن، از راه‌اندازی محیط توسعه‌ی محلی (با استفاده از Laragon) تا آماده‌سازی نهایی برای انتشار، به صورت گام به گام پیاده‌سازی و مستند گردید. اقدامات کلیدی انجام‌شده شامل بومی‌سازی و شمس‌سازی کامل سیستم‌های تاریخ‌نگار (با جایگزینی daterangepicker و moment با نسخه‌های جلالی و ادغام کتابخانه‌ی jdatetime در منطق PHP)، راست‌چین‌سازی کامل رابط کاربری با اعمال تغییرات گسترده در CSS و افزودن فونت‌های فارسی، ترجمه‌ی جامع تمامی متون و پست‌تایپ‌ها، و همچنین بررسی و توسعه‌ی راهکارهایی برای یکپارچه‌سازی درگاه‌های پرداخت بومی (مانند IDPay) و مدیریت چالش‌های لایسنس بین‌المللی بود.

نتایج این تحقیق نشان داد که فرآیند بومی‌سازی، فراتر از ترجمه صرف بوده و نیازمند دانش فنی عمیق در زمینه‌های برنامه‌نویسی سمت کلاینت و سرور (JavaScript, PHP, CSS) و همچنین شناخت دقیق از ساختار وردپرس و نیازهای بومی کشور است. چالش‌های عمده‌ای نظیر پیچیدگی یکپارچه‌سازی تقویم شمسی در هر دو سمت فرانت‌اند و بک‌اند، نیاز به تغییرات گسترده در CSS برای RTL صحیح، و عدم رعایت کامل Internationalization در برخی محصولات بین‌المللی شناسایی و راهکارهایی برای آن‌ها ارائه شد.

این پژوهش، با ارائه‌ی یک راهنمای عملی و مستندسازی دقیق چالش‌ها، به توسعه‌دهندگان ایرانی در تولید محصولات وردپرسی با کیفیت و مطابق با استانداردهای مارکت‌پلیس‌های داخلی (نظیر ژاکت و راست‌چین) کمک می‌کند. پیشنهاداتی برای کارهای آتی شامل خودکارسازی فرآیند بومی‌سازی، توسعه‌ی قالب‌ها و افزونه‌های بومی از ابتدا، و بهبود همکاری با مارکت‌پلیس‌ها ارائه شده است تا به غنی‌سازی هرچه بیشتر اکوسیستم وردپرس فارسی و افزایش دسترسی کاربران به ابزارهای کارآمد کمک شود.

فصل اول: کلیات تحقیق

۱-۱) مقدمه

در دنیای امروز که فناوری اطلاعات و ارتباطات با سرعت سرسام‌آوری در حال پیشرفت است، وبسایت‌ها به بخش جدایی‌ناپذیری از زندگی روزمره، کسب‌وکارها، و فعالیت‌های اجتماعی تبدیل شده‌اند. از میان پلتفرم‌های مدیریت محتوا (CMS) موجود، وردپرس (WordPress) به دلیل انعطاف‌پذیری بالا، جامعه کاربری گسترده، و قابلیت‌های فراوان، به یکی از محبوب‌ترین و پرکاربردترین سیستم‌های مدیریت محتوا در سطح جهان تبدیل شده است. این CMS متن‌باز، امکان ایجاد انواع وبسایت‌ها از وبلاگ‌های شخصی گرفته تا فروشگاه‌های آنلاین پیچیده و وبسایت‌های شرکتی را فراهم می‌آورد.

با توجه به گسترش روزافزون استفاده از وردپرس در ایران، نیاز به بومی‌سازی (Localization) قالب‌ها (Themes) و افزونه‌ها (Plugins) بیش از پیش احساس می‌شود. بومی‌سازی فرآیندی فراتر از صرفاً ترجمه متون است؛ این فرآیند شامل تطبیق جنبه‌های مختلف یک نرم‌افزار یا وبسایت با فرهنگ، زبان و نیازهای خاص یک منطقه جغرافیایی یا کشور است. در مورد وردپرس، بومی‌سازی شامل فارسی‌سازی متون، راست‌چین‌سازی (RTL) رابط کاربری برای زبان فارسی، تطبیق فرمت‌های تاریخ و زمان (تقویم شمسی) و همچنین یکپارچه‌سازی با درگاه‌های پرداخت محلی و سایر خدمات بومی می‌شود.

مارکت پلیس‌های ایرانی نظیر ژاکت و راست‌چین، نقش حیاتی در فراهم آوردن قالب‌ها و افزونه‌های بومی‌سازی شده برای کاربران ایرانی ایفا می‌کنند. این پلتفرم‌ها به توسعه‌دهندگان ایرانی این امکان را می‌دهند که محصولات بومی‌سازی شده خود را عرضه کنند و به این ترتیب، دسترسی کاربران به ابزارهای مناسب و کاربرپسند را فراهم می‌آورند. با این حال، فرآیند بومی‌سازی و ارائه محصولات در این مارکت پلیس‌ها چالش‌ها و پیچیدگی‌های خاص خود را دارد که نیاز به بررسی دقیق و ارائه راهکارهای عملی دارد.

این پروژه با هدف بررسی جامع فرآیند بومی‌سازی قالب‌ها و افزونه‌های وردپرس برای مارکت پلیس‌های ایرانی، به شناسایی چالش‌ها و ارائه راهکارهای عملی در این زمینه می‌پردازد. امید است نتایج این تحقیق بتواند به توسعه‌دهندگان، طراحان وب، و فعالان حوزه وردپرس در ایران کمک کند تا محصولات با کیفیت‌تر و کاربرپسندتری را ارائه دهند و سهمی در غنی‌سازی اکوسیستم وردپرس فارسی داشته باشد.

۱-۲) بیان مسأله

گسترش روزافزون اینترنت و نفوذ فناوری در ابعاد مختلف زندگی، باعث شده است که کسب‌وکارهای سنتی نیز به سمت حضور آنلاین سوق پیدا کنند. در این میان، وبسایت‌ها به عنوان مهم‌ترین ابزار حضور آنلاین،

نقش کلیدی ایفا می‌کنند. وردپرس، به عنوان یک سیستم مدیریت محتوای محبوب و قدرتمند، به دلیل سادگی استفاده و انعطاف‌پذیری بالا، انتخاب اول بسیاری از کاربران و کسب‌وکارها در سراسر جهان و به خصوص در ایران است. با وجود قابلیت‌های فراوان و جامعه توسعه‌دهنده فعال، بسیاری از قالب‌ها و افزونه‌های وردپرس در ابتدا برای مخاطبان بین‌المللی و با زبان انگلیسی طراحی می‌شوند. این امر منجر به بروز چالش‌های جدی برای کاربران فارسی‌زبان می‌شود که مهم‌ترین آن‌ها عدم تطابق فرهنگی و زبانی است.

عدم بومی‌سازی مناسب این محصولات، مشکلاتی نظیر موارد زیر را به وجود می‌آورد:

- **ناخوانایی و مشکلات ظاهری:** متون انگلیسی در وبسایت‌های فارسی‌زبان، به دلیل عدم پشتیبانی کامل از زبان فارسی و راست‌چین‌سازی (RTL)، می‌توانند ظاهر نامناسبی داشته باشند. فونت‌ها، چینش عناصر، و تراز متون ممکن است به درستی نمایش داده نشوند.
- **تجربه کاربری نامناسب:** عدم ترجمه صحیح و کامل پنل مدیریت، پیغام‌ها و راهنماها، باعث سردرگمی کاربران و کاهش کارایی وبسایت می‌شود. کاربران فارسی‌زبان انتظار دارند با یک محیط کاملاً فارسی و منطبق با فرهنگ خود مواجه باشند.
- **ناتوانی در استفاده از قابلیت‌های محلی:** بسیاری از قالب‌ها و افزونه‌ها با سیستم‌های تاریخ و زمان میلادی کار می‌کنند. برای کاربران ایرانی، استفاده از تقویم شمسی و فرمت‌های تاریخ بومی از اهمیت بالایی برخوردار است. همچنین، یکپارچه‌سازی با درگاه‌های پرداخت بانکی ایرانی و سایر خدمات مالی بومی، برای کسب‌وکارهای آنلاین حیاتی است.
- **محدودیت‌های فنی و قانونی:** برخی از سرویس‌ها یا API های بین‌المللی ممکن است به دلیل تحریم‌ها یا محدودیت‌های جغرافیایی برای کاربران ایرانی قابل دسترس نباشند. این امر نیاز به جایگزینی یا توسعه راه‌حل‌های بومی را ایجاد می‌کند.
- **چالش‌های توسعه‌دهندگان:** فرآیند بومی‌سازی دستی و بدون استانداردسازی مشخص، زمان‌بر، پرهزینه و مستعد خطا است. توسعه‌دهندگان ایرانی که قصد دارند قالب‌ها و افزونه‌های بومی‌سازی شده را در مارکت‌پلیس‌های داخلی عرضه کنند، با نبود یک راهنمای جامع و یکپارچه برای انجام این فرآیند مواجه هستند.

مارکت‌پلیس‌های ایرانی مانند ژاکت و راست‌چین، تلاش می‌کنند با ارائه محصولات بومی‌سازی شده، این نیازها را برطرف کنند. اما خود فرآیند ارائه محصول در این مارکت‌پلیس‌ها نیز شامل مراحل از آماده‌سازی، تست و اخذ تاییدیه است که نیازمند درک دقیق و رعایت اصول خاصی است. این پروژه با بررسی جامع فرآیند

بومی سازی و ارائه در مارکت پلیس های ایرانی، به دنبال پاسخ به این چالش ها و ارائه یک مدل عملیاتی برای بهبود کیفیت و سرعت این فرآیند است.

۱-۳) اهمیت و ضرورت تحقیق

اهمیت و ضرورت این تحقیق را می توان از چندین منظر مورد بررسی قرار داد:

- پوشش نیازهای کاربران فارسی زبان: با توجه به رشد چشمگیر استفاده از وردپرس در ایران، میلیون ها کاربر فارسی زبان نیازمند قالب ها و افزونه هایی هستند که کاملاً با زبان و فرهنگ آنها سازگار باشد. این سازگاری نه تنها شامل ترجمه متون، بلکه شامل پشتیبانی از تقویم شمسی، راست چین سازی دقیق و یکپارچگی با سیستم های پرداخت بومی است. این تحقیق با ارائه یک نقشه راه برای بومی سازی، به ارتقاء تجربه کاربری این بخش بزرگ از کاربران کمک می کند.
- توانمندسازی توسعه دهندگان داخلی: بسیاری از توسعه دهندگان و وب مستران ایرانی علاقه دارند قالب ها و افزونه های وردپرس را بومی سازی کرده و در مارکت پلیس های داخلی عرضه کنند. این تحقیق با مستندسازی فرآیندها، ابزارها و چالش های موجود، یک منبع ارزشمند برای این افراد خواهد بود. ارائه کدهای نمونه و راهنمایی های عملی، می تواند به کاهش زمان و هزینه بومی سازی کمک کند و کیفیت محصولات نهایی را بهبود بخشد.
- تقویت اکوسیستم وردپرس فارسی: با تسهیل فرآیند بومی سازی، تعداد و کیفیت محصولات فارسی سازی شده در مارکت پلیس های ایرانی افزایش خواهد یافت. این امر به نوبه خود، به رشد و توسعه اکوسیستم وردپرس فارسی و افزایش رقابت پذیری در این حوزه کمک شایانی خواهد کرد.
- افزایش درآمدزایی و فرصت های شغلی: ارائه محصولات بومی سازی شده با کیفیت بالا در مارکت پلیس ها، می تواند به ایجاد فرصت های شغلی جدید برای توسعه دهندگان، مترجمان، و متخصصان وردپرس منجر شود. همچنین، این امر به افزایش درآمدزایی برای مارکت پلیس ها و توسعه دهندگان محصولات کمک می کند.
- غنی سازی محتوای بومی: بومی سازی صحیح قالب ها و افزونه ها، نه تنها به استفاده راحت تر از آنها کمک می کند، بلکه باعث غنی تر شدن محتوای بومی در فضای وب فارسی می شود. این شامل تولید محتوای آموزشی، مستندات، و پشتیبانی به زبان فارسی برای محصولات بومی سازی شده است.
- مقابله با چالش های تحریم: با توجه به محدودیت ها و تحریم های بین المللی، دسترسی به برخی از ابزارها، سرویس ها، و لایسنس های بین المللی برای کاربران ایرانی دشوار است. این تحقیق با بررسی

راه‌حل‌های بومی و جایگزین، به توسعه‌دهندگان کمک می‌کند تا محصولات خود را بدون وابستگی به سرویس‌های تحریم‌شده ارائه دهند.

در مجموع، این پروژه نه تنها به حل یک مشکل فنی در حوزه وردپرس می‌پردازد، بلکه ابعاد اقتصادی، فرهنگی، و اجتماعی گسترده‌ای دارد و می‌تواند به پیشرفت اکوسیستم دیجیتال ایران کمک کند.

۴-۱) اهداف تحقیق

اهداف اصلی این تحقیق به شرح زیر است:

۱-۴-۱) هدف اصلی:

- بررسی جامع و ارائه یک مدل عملیاتی برای فرآیند بومی‌سازی قالب‌ها و افزونه‌های وردپرس جهت عرضه در مارکت پلیس‌های ایرانی.

۲-۴-۱) اهداف فرعی:

- شناسایی مراحل بومی‌سازی: شناسایی و تشریح گام به گام مراحل فنی و غیرفنی لازم برای بومی‌سازی کامل یک قالب یا افزونه وردپرس، شامل ترجمه، راست‌چین‌سازی و تطبیق با استانداردهای بومی.
- معرفی ابزارها و تکنیک‌ها: معرفی و بررسی کاربردی ابزارها و تکنیک‌های مؤثر در فرآیند بومی‌سازی، از جمله ابزارهای ترجمه (مانند Poedit و Loco Translate)، ویرایشگرهای کد و روش‌های بهینه‌سازی CSS برای RTL.
- تبیین چالش‌ها و راهکارها: شناسایی چالش‌های رایج و مشکلات فنی که توسعه‌دهندگان در فرآیند بومی‌سازی و ارائه محصول در مارکت پلیس‌های ایرانی با آن‌ها مواجه می‌شوند (مانند مشکلات تقویم شمسی، درگاه پرداخت، و لایسنس‌گذاری) و ارائه راهکارهای عملی برای غلبه بر آن‌ها.
- تطبیق با نیازهای بومی: بررسی و ارائه روش‌هایی برای تطبیق قابلیت‌های بین‌المللی قالب‌ها و افزونه‌ها (مانند سیستم تاریخ و زمان) با نیازهای خاص کاربران ایرانی (مانند تقویم شمسی و درگاه‌های پرداخت بومی).

- **مستندسازی فرآیند عرضه:** مستندسازی فرآیند و الزامات لازم برای عرضه محصولات بومی سازی شده در مارکت پلیس های ایرانی (مانند ژاکت و راست چین)، از جمله آماده سازی فایل ها، ایجاد پیش نمایش، و رعایت استانداردهای محتوایی.
- **ارائه نمونه عملی:** انجام یک مورد کاوی عملی بومی سازی یک قالب مطرح بین المللی وردپرس و مستندسازی تمامی مراحل و تجربیات به دست آمده، به عنوان یک راهنمای عملی برای سایر توسعه دهندگان.

۵-۱) سوالات تحقیق

این تحقیق به دنبال پاسخگویی به سؤالات زیر است:

۱-۵-۱) سؤال اصلی:

- فرآیند بومی سازی قالب ها و افزونه های وردپرس برای عرضه در مارکت پلیس های ایرانی شامل چه مرحله ای است و چه چالش ها و راهکارهایی در این مسیر وجود دارد؟

۱-۵-۲) سؤالات فرعی:

- مراحل اصلی فرآیند بومی سازی یک قالب یا افزونه وردپرس برای زبان فارسی و محیط ایران کدامند؟
- چه ابزارها و تکنیک هایی برای انجام هر یک از این مراحل بومی سازی (ترجمه، راست چین سازی، تطبیق تقویم، یکپارچه سازی پرداخت) کارآمدتر هستند؟
- چالش های فنی و غیرفنی رایج در بومی سازی محصولات وردپرسی برای مارکت پلیس های ایرانی چیست و چگونه می توان بر آنها فائق آمد؟
- الزامات و مراحل عرضه یک محصول بومی سازی شده در مارکت پلیس های ایرانی نظیر ژاکت و راست چین کدامند؟
- چه راهکارهایی برای بهبود و خودکارسازی فرآیند بومی سازی و افزایش کیفیت محصولات عرضه شده در آینده پیشنهاد می شود؟

۱-۶) روش تحقیق

برای دستیابی به اهداف تحقیق و پاسخگویی به سؤالات مطرح شده، از یک رویکرد ترکیبی شامل روش‌های کیفی و کمی، با تمرکز بر روش‌های عملی و مطالعه موردی (Case Study) استفاده خواهد شد. این روش شامل مراحل زیر است:

- **مطالعه و بررسی ادبیات:** در این مرحله، به بررسی جامع متون علمی، مقالات، کتاب‌ها و مستندات مربوط به سیستم مدیریت محتوای وردپرس، مفهوم بومی‌سازی (Localization) در نرم‌افزار و وب، ساختار قالب‌ها و افزونه‌های وردپرس، و همچنین مقالات و پروژه‌های مشابه در زمینه بومی‌سازی وردپرس پرداخته خواهد شد. هدف از این بخش، ایجاد یک چارچوب نظری قوی و شناسایی بهترین شیوه‌ها (Best Practices) در این حوزه است.
- **تحلیل و بررسی مارکت پلیس‌های ایرانی:** در این بخش، مارکت پلیس‌های فعال در حوزه وردپرس در ایران، نظیر ژاکت و راست‌چین، مورد بررسی قرار می‌گیرند. این بررسی شامل مدل کسب‌وکار، الزامات فنی و محتوایی برای پذیرش محصولات و نحوه تعامل با توسعه‌دهندگان است.
- **مطالعه موردی (Case Study) عملی:** بخش عمده‌ای از این تحقیق بر پایه یک مطالعه موردی عملی استوار خواهد بود. در این مرحله، یک قالب مطرح بین‌المللی از ThemeForest (مانند قالب "تراولر") انتخاب شده و فرآیند کامل بومی‌سازی آن به صورت عملی انجام می‌شود. مراحل این مطالعه موردی شامل:
 - **انتخاب و خرید قالب:** شامل بررسی نکات مهم قبل از خرید و فرآیند خرید قالب.
 - **راه‌اندازی محیط توسعه محلی:** نصب و پیکربندی ابزارهای لازم مانند Laragon (یا XAMPP/WAMP) و وردپرس در یک محیط محلی.
 - **نصب قالب و افزونه‌های مورد نیاز:** نصب قالب انتخاب شده و افزونه‌های همراه آن.
 - **بررسی و تحلیل ساختار قالب:** شناسایی فایل‌های اصلی، ساختار کد و بخش‌های قابل بومی‌سازی (متون، تاریخ‌ها، استایل‌ها).
 - **ترجمه متون:** استفاده از ابزارهایی مانند Poedit یا Loco Translate برای استخراج رشته‌های متنی و ترجمه آن‌ها به فارسی.
 - **راست‌چین‌سازی (RTL):** اعمال تغییرات لازم در فایل‌های CSS و HTML برای پشتیبانی کامل از زبان فارسی (RTL)، شامل تنظیمات فونت، تراز متون و چینش عناصر.

- **بومی سازی تقویم و زمان:** جایگزینی سیستم های تاریخ و زمان میلادی با تقویم شمسی (مثلاً با استفاده از کتابخانه هایی مانند `zdatetime` و `daterangepicker` جلالی).
- **یکپارچه سازی درگاه پرداخت بومی:** بررسی و توسعه راهکارهایی برای اتصال قالب به درگاه های پرداخت ایرانی (مانند `IDPay`)
- **رفع مشکلات لایسنس بین الملل:** بررسی و ارائه راهکارهایی برای مدیریت تحریم ها و مشکلات مربوط به لایسنس محصولات بین المللی.
- **تست و اشکال زدایی (Debugging):** استفاده از ابزارهایی مانند `Composer` و `dd` برای دیباگ کردن و رفع مشکلات فنی.
- **مستندسازی فرآیند:** ثبت دقیق تمامی مراحل، ابزارهای استفاده شده، چالش های مواجه شده، و راهکارهای اتخاذ شده در طول فرآیند بومی سازی.
- **تحلیل یافته ها و ارائه راهکارها:** پس از اتمام مطالعه موردی و مستندسازی، داده های جمع آوری شده مورد تجزیه و تحلیل قرار می گیرند. چالش های رایج شناسایی شده و راهکارهای عملی برای غلبه بر آنها پیشنهاد می شود.
- **نتیجه گیری و پیشنهادات:** در نهایت، نتایج تحقیق جمع بندی شده و پیشنهادات برای تحقیقات آتی و بهبود فرآیند بومی سازی در اکوسیستم وردپرس ایران ارائه خواهد شد.

۷-۱) تعریف واژگان کلیدی

۱. **وردپرس (WordPress):** یک سیستم مدیریت محتوای (CMS) متن باز و رایگان است که برای ساخت و مدیریت وبسایت ها و وبلاگ ها استفاده می شود.
۲. **بومی سازی (Localization - L10n):** فرآیند تطبیق یک محصول یا خدمات (مانند نرم افزار یا وبسایت) با یک زبان، فرهنگ، و منطقه ی خاص. این فرآیند فراتر از صرفاً ترجمه است و شامل تغییرات فنی و فرهنگی می شود.
۳. **قالب (Theme):** مجموعه ای از فایل ها (شامل کدها، استایل ها، و تصاویر) که ظاهر و طرح بندی یک وبسایت وردپرس را تعیین می کند.
۴. **افزونه (Plugin):** یک برنامه نرم افزاری کوچک که قابلیت های جدیدی را به وبسایت وردپرس اضافه می کند یا قابلیت های موجود را گسترش می دهد.

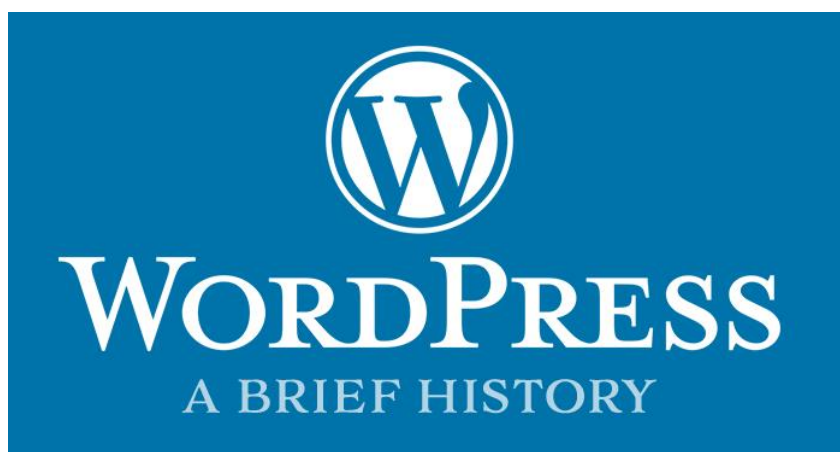
۵. **مارکت پلیس (Marketplace):** یک پلتفرم آنلاین که در آن فروشندگان (توسعه دهندگان) می توانند محصولات دیجیتالی خود (مانند قالب ها و افزونه های وردپرس) را به خریداران عرضه کنند. در ایران، ژاکت و راست چین از جمله این مارکت پلیس ها هستند.
۶. **راست چین سازی (Right-to-Left - RTL):** فرآیند تطبیق رابط کاربری و نمایش متن برای زبان هایی که از راست به چپ نوشته می شوند (مانند فارسی و عربی). این شامل تغییرات در CSS و ساختار HTML است.
۷. **ThemeForest:** یکی از بزرگترین بازارهای آنلاین بین المللی برای خرید و فروش قالب های وردپرس و سایر سیستم های مدیریت محتوا.
۸. **ژاکت (Zhaket):** یکی از مارکت پلیس های بزرگ ایرانی برای عرضه قالب ها و افزونه های وردپرس بومی سازی شده.
۹. **راست چین:** یکی دیگر از مارکت پلیس های محبوب ایرانی در زمینه وردپرس که محصولات بومی سازی شده را ارائه می دهد.
۱۰. **واژه XAMPP/WAMP/Larago:** بسته های نرم افزاری رایگان و متن باز که امکان راه اندازی یک وب سرور محلی (Local Server) با Apache، MySQL و PHP را فراهم می کنند تا بتوان وب سایت های وردپرسی را به صورت آفلاین توسعه داد و تست کرد.
۱۱. **Poedit:** یک نرم افزار ویرایشگر برای فایل های .po و .mo که برای ترجمه برنامه ها و وب سایت ها استفاده می شود.
۱۲. **Loco Translate:** یک افزونه وردپرس که امکان ترجمه قالب ها و افزونه ها را به صورت مستقیم در پنل مدیریت وردپرس فراهم می کند.
۱۳. **jdatetime:** یک کتابخانه PHP برای کار با تاریخ های شمسی (جلالی).
۱۴. **daterangepicker:** یک کتابخانه جاوا اسکریپت برای انتخاب محدوده تاریخ.

فصل ۲: مروری بر ادبیات تحقیق و پیشینه تحقیق

در این فصل، به بررسی مبانی نظری و پیشینه‌ی تحقیقاتی مرتبط با موضوع بومی‌سازی قالب‌ها و افزونه‌های وردپرس می‌پردازیم. این بررسی شامل معرفی سیستم مدیریت محتوای وردپرس، تشریح ساختار قالب‌ها و افزونه‌ها، تبیین مفهوم بومی‌سازی و اهمیت آن در فضای وب، معرفی مارکت‌پلیس‌های ایرانی و در نهایت، مرور مقالات و پروژه‌های مشابه انجام‌شده در این حوزه است. هدف این فصل، فراهم آوردن یک بستر دانش نظری برای درک عمیق‌تر فرآیندهای عملیاتی مطرح‌شده در فصول بعدی است.

۲-۱) معرفی CMS وردپرس

سیستم مدیریت محتوا (Content Management System - CMS) نرم‌افزاری است که به کاربران امکان می‌دهد بدون نیاز به دانش برنامه‌نویسی پیشرفته، محتوای وبسایت خود را ایجاد، ویرایش، مدیریت و منتشر کنند. CMS ها فرآیند طراحی وبسایت را تسهیل کرده و نگهداری آن را ساده‌تر می‌سازند. در میان انواع بی‌شمار CMS های موجود، وردپرس (WordPress) به عنوان یکی از محبوب‌ترین و پرکاربردترین سیستم‌ها در سطح جهان شناخته می‌شود.



تصویر ۲-۱: لوگوی وردپرس

۲-۱-۱) تاریخچه و آمار استفاده از وردپرس

وردپرس در سال ۲۰۰۳ توسط مت مولنوک (Matt Mullenweg) و مایک لیتل (Mike Little) به عنوان شاخه‌ای از پروژه B2/cafelog آغاز به کار کرد. هدف اولیه از ایجاد وردپرس، فراهم آوردن یک پلتفرم ساده و کارآمد برای وبلاگ‌نویسی بود. اما با گذشت زمان و توسعه‌ی مستمر، این سیستم فراتر از یک ابزار وبلاگ‌نویسی رفت و به یک CMS قدرتمند و چندمنظوره تبدیل شد. قابلیت‌های افزوده، معماری قابل توسعه، و جامعه‌ی کاربری و توسعه‌دهنده‌ی فعال، وردپرس را به انتخابی ایده‌آل برای ساخت انواع وبسایت‌ها تبدیل کرده است.

بر اساس آمارهای اخیر، وردپرس بیش از ۴۳ درصد از کل وبسایت‌های فعال در اینترنت را قدرت می‌بخشد. این آمار نشان‌دهنده تسلط بلامنازع وردپرس بر بازار CMS ها است و آن را به پرکاربردترین پلتفرم برای ساخت وبسایت‌های شخصی، شرکتی، فروشگاه‌ها، خبری و بسیاری دیگر تبدیل کرده است. این گستردگی استفاده، به دلیل ویژگی‌هایی نظیر:

- **متن‌باز و رایگان بودن:** هسته‌ی وردپرس به صورت متن‌باز و رایگان در دسترس است که هزینه‌های اولیه توسعه وبسایت را به شدت کاهش می‌دهد.
- **سهولت استفاده:** رابط کاربری ساده و بصری وردپرس، حتی برای کاربران غیرفنی نیز قابل‌درک و استفاده است.
- **انعطاف‌پذیری بالا:** با هزاران قالب و افزونه، وردپرس قابلیت سفارشی‌سازی بی‌نهایتی را برای رفع نیازهای مختلف فراهم می‌آورد.
- **جامعه‌ی بزرگ و فعال:** وجود یک جامعه‌ی عظیم از توسعه‌دهندگان، طراحان و کاربران، به معنای دسترسی آسان به پشتیبانی، مستندات و منابع آموزشی فراوان است.
- **سازگار با سئو (SEO-Friendly):** وردپرس به طور پیش‌فرض برای موتورهای جستجو بهینه‌سازی شده است و افزونه‌های متعددی نیز برای بهبود سئو در دسترس هستند.

۲-۱-۲) مزایا و معایب استفاده از وردپرس

هر سیستمی، در کنار مزایای فراوان، دارای معایبی نیز هست که شناخت آن‌ها برای انتخاب آگاهانه حائز اهمیت است.

مزایا:

انعطاف‌پذیری و قابلیت توسعه: از طریق قالب‌ها (Themes) و افزونه‌ها (Plugins)، وردپرس قابلیت‌های بی‌شماری را ارائه می‌دهد که از یک وبلاگ ساده تا یک فروشگاه آنلاین پیچیده یا یک شبکه‌ی اجتماعی قابل گسترش است.

- **سهولت یادگیری و استفاده:** رابط کاربری دوستانه و وجود منابع آموزشی فراوان، یادگیری و کار با وردپرس را برای افراد با سطوح دانش فنی مختلف آسان می‌سازد.

- **متن باز بودن و هزینه‌ی پایین:** هسته‌ی وردپرس رایگان است و همین امر باعث می‌شود هزینه‌های راه‌اندازی اولیه وبسایت به مراتب کمتر از سیستم‌های اختصاصی باشد.
- **پشتیبانی قوی جامعه کاربری:** در صورت بروز مشکل یا نیاز به راهنمایی، می‌توان به راحتی از طریق انجمن‌ها، گروه‌های آنلاین و مستندات رسمی، پاسخ سوالات را یافت.
- **بهینه‌سازی برای موتورهای جستجو (SEO):** وردپرس به صورت ذاتی ساختار مناسبی برای سئو دارد و افزونه‌های قدرتمندی نیز برای بهبود رتبه در موتورهای جستجو ارائه می‌دهد.
- **امنیت (با رعایت نکات):** با وجود برخی نگرانی‌ها، وردپرس در صورت به‌روزرسانی منظم و استفاده از افزونه‌های امنیتی معتبر، از امنیت بالایی برخوردار است.

معایب:

- **نیاز به به‌روزرسانی مداوم:** برای حفظ امنیت و کارایی، هسته‌ی وردپرس، قالب‌ها و افزونه‌ها باید به طور منظم به‌روزرسانی شوند.
- **سرعت و عملکرد (در صورت عدم بهینه‌سازی):** در صورت نصب تعداد زیادی افزونه، استفاده از قالب‌های سنگین و عدم بهینه‌سازی صحیح، ممکن است سرعت وبسایت کاهش یابد.
- **مسائل امنیتی (در صورت عدم رعایت نکات):** اگر وردپرس به درستی پیکربندی نشود، یا از افزونه‌ها و قالب‌های نامعتبر استفاده شود، ممکن است در معرض حملات امنیتی قرار گیرد.
- **پیچیدگی در سفارشی‌سازی‌های عمیق:** برای انجام تغییرات بسیار خاص و پیچیده در کد اصلی، نیاز به دانش برنامه‌نویسی PHP، HTML، CSS و JavaScript است.
- **پشتیبانی محدود از برخی ویژگی‌های بومی:** قالب‌ها و افزونه‌های طراحی‌شده برای بازارهای بین‌المللی ممکن است به صورت پیش‌فرض از تقویم شمسی، درگاه‌های پرداخت ایرانی و راست‌چین‌سازی کامل پشتیبانی نکنند که نیاز به بومی‌سازی دارد.

با توجه به مزایا و معایب مطرح‌شده، وردپرس همچنان یک گزینه قدرتمند و اقتصادی برای ایجاد انواع وبسایت‌ها محسوب می‌شود، به خصوص با توجه به قابلیت بومی‌سازی که موضوع اصلی این پایان‌نامه است.

۲-۲) ساختار قالب‌ها و پلاگین‌ها در وردپرس

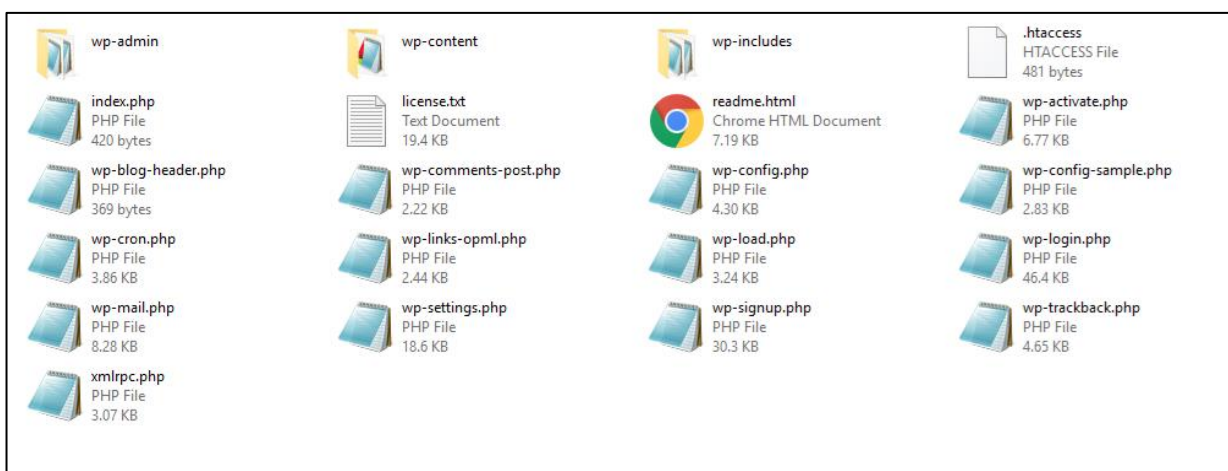
درک ساختار قالب‌ها و افزونه‌ها در وردپرس برای هرگونه فعالیت بومی‌سازی یا توسعه‌ای ضروری است. این دو عنصر، پایه و اساس قابلیت توسعه‌پذیری و انعطاف‌پذیری وردپرس را تشکیل می‌دهند.

۲-۲-۱) ساختار فایل‌های قالب وردپرس

قالب وردپرس مجموعه‌ای از فایل‌هاست که ظاهر و عملکرد وبسایت شما را تعیین می‌کند. هر قالب حداقل شامل دو فایل اصلی است `style.css` (برای استایل‌دهی) و `index.php` (برای نمایش محتوا). با این حال، قالب‌های مدرن و پیچیده‌تر از ساختار سلسله‌مراتبی گسترده‌تری از فایل‌ها و پوشه‌ها پیروی می‌کنند. در یک قالب استاندارد وردپرس، فایل‌های زیر به طور معمول یافت می‌شوند:

- **style.css**: این فایل حاوی اطلاعات اصلی قالب مانند نام، نویسنده، نسخه، و توضیحات است. همچنین، شامل تمامی کدهای CSS برای استایل‌دهی به عناصر وبسایت است.
- **index.php**: فایل اصلی قالب است که به عنوان یک فایل fallback (جایگزین) عمل می‌کند. در صورتی که فایل‌های خاص‌تری برای نمایش صفحات مختلف (مانند `home.php` یا `single.php`) وجود نداشته باشند، وردپرس از این فایل استفاده می‌کند.
- **functions.php**: این فایل به عنوان یک قلاب (hook) برای افزودن قابلیت‌های سفارشی به قالب عمل می‌کند. از طریق این فایل می‌توان توابع جدید تعریف کرد، قابلیت‌های پیش‌فرض وردپرس را تغییر داد و یا تنظیمات خاصی را برای قالب اعمال کرد.
- **header.php**: بخشی از قالب که شامل عناصر `<head>` و ابتدای بدنه‌ی سند HTML است، از جمله عنوان صفحه، متا تگ‌ها، لینک به فایل‌های CSS و JavaScript و شروع منوها.
- **footer.php**: بخشی از قالب که شامل پایان بدنه‌ی سند HTML و عناصر پایینی صفحه مانند کپی‌رایت و کدهای جاوااسکریپت انتهایی است.
- **sidebar.php**: برای نمایش سایدبار (نوار کناری) وبسایت استفاده می‌شود. این فایل معمولاً شامل ویجت‌ها و ناحیه‌های ابزارک است.
- **single.php**: مسئول نمایش پست‌های تکی (مقاله‌ها یا نوشته‌ها) است.
- **page.php**: مسئول نمایش صفحات استاتیک (مانند درباره ما، تماس با ما) است.
- **archive.php**: برای نمایش صفحات آرشیو (مانند دسته‌بندی‌ها، برچسب‌ها، نویسندگان، تاریخ‌ها) استفاده می‌شود.
- **search.php**: مسئول نمایش نتایج جستجو است.
- **comments.php**: برای نمایش فرم نظرات و لیست نظرات یک پست یا صفحه.
- **404.php**: صفحه خطای ۴۰۴ (Not Found) را نمایش می‌دهد.

- **screenshot.png**: یک فایل تصویری (معمولاً ۱۲۰۰ در ۹۰۰ پیکسل) که در پنل مدیریت وردپرس به عنوان تصویر پیش‌نمایش قالب نمایش داده می‌شود.
- پوشه‌ی **inc/** یا **includes/**: شامل فایل‌های PHP کوچک‌تر برای سازماندهی بهتر کد، مانند توابع کمکی، کلاس‌ها، یا تعریف پست‌تایپ‌های سفارشی.
- پوشه‌ی **assets/** یا **css/**، **js/**، **images/**: شامل فایل‌های CSS، JavaScript و تصاویر مورد نیاز قالب.
- پوشه‌ی **languages/**: این پوشه برای نگهداری فایل‌های ترجمه (با پسوندهای .po و .mp) قالب است که در فرآیند بومی‌سازی اهمیت زیادی دارد.



تصویر ۲-۲: ساختار فایل‌های وردپرس

۲-۲-۲) ساختار فایل‌های افزونه وردپرس

افزونه‌های وردپرس، قابلیت‌های جدیدی را به هسته‌ی وردپرس اضافه می‌کنند یا عملکرد آن را تغییر می‌دهند. یک افزونه حداقل شامل یک فایل PHP اصلی است که در بالای آن، اطلاعات مربوط به افزونه (نام، توضیحات، نویسنده، نسخه و...) با استفاده از کامنت‌های PHP قرار می‌گیرد. افزونه‌های پیچیده‌تر، ساختار پوشه‌ای سازمان‌یافته‌ای دارند. ساختار کلی یک افزونه می‌تواند شامل موارد زیر باشد:

- **فایل PHP اصلی افزونه**: این فایل (مثلاً my-plugin.php) نقطه شروع افزونه است و شامل اطلاعات هدر افزونه و کدهای اصلی برای راه‌اندازی قابلیت‌هاست.
- پوشه‌ی **admin/**: شامل فایل‌های مربوط به رابط کاربری مدیریت افزونه در پنل ادمین وردپرس.
- پوشه‌ی **public/**: شامل فایل‌هایی که در سمت کاربر (Frontend) نمایش داده می‌شوند.

- پوشه‌ی **includes/** یا **inc/**: شامل توابع و کلاس‌های کمکی برای سازماندهی بهتر کد.
- پوشه‌ی **assets/**: شامل فایل‌های CSS، JavaScript و تصاویر افزونه.
- پوشه‌ی **languages/**: مشابه قالب‌ها، این پوشه نیز برای نگهداری فایل‌های ترجمه افزونه (po) و (mo) استفاده می‌شود.
- پوشه‌ی **templates/**: در برخی افزونه‌ها، فایل‌های قالب‌بندی (Template) برای نمایش بخش‌های خاصی از افزونه در این پوشه قرار می‌گیرند.
- **uninstall.php**: فایلی که حاوی کدهایی برای پاک‌سازی کامل اطلاعات افزونه هنگام حذف آن از وردپرس است.

درک این ساختارها برای بومی‌سازی حیاتی است، زیرا بسیاری از رشته‌های متنی برای ترجمه، در فایل‌های PHP و قالب‌ها قرار دارند و استایل‌ها نیاز به تغییر برای پشتیبانی از RTL دارند.

۲-۳) مفهوم بومی‌سازی (Localization) در نرم‌افزار و وب

بومی‌سازی (Localization) که با نماد L10n نیز نمایش داده می‌شود (به دلیل ۱۰ حرف بین L و n)، فرآیندی جامع‌تر از صرفاً ترجمه است. در واقع، ترجمه تنها یکی از مراحل بومی‌سازی محسوب می‌شود. بومی‌سازی به معنای تطبیق یک محصول (نرم‌افزار، وب‌سایت، بازی و...) با زبان، فرهنگ، و نیازهای خاص یک منطقه جغرافیایی یا کشور خاص است. هدف نهایی بومی‌سازی، ایجاد یک تجربه کاربری بومی و طبیعی برای مخاطبان هدف است، به گونه‌ای که محصول به نظر نرسد از ابتدا برای آن منطقه طراحی نشده است.

۲-۳-۱) تعریف بومی‌سازی و تمایز آن با ترجمه (Translation) و جهانی‌سازی (Globalization)

برای درک بهتر مفهوم بومی‌سازی، لازم است آن را از دو مفهوم مرتبط دیگر، یعنی ترجمه و جهانی‌سازی، تمایز دهیم:

- **ترجمه (Translation):** صرفاً فرآیند تبدیل متن از یک زبان به زبان دیگر است، با حفظ معنا و مفهوم اصلی. ترجمه یک جزء ضروری از بومی‌سازی است، اما به تنهایی کافی نیست. برای مثال، ترجمه یک وب‌سایت از انگلیسی به فارسی تنها شامل تغییر کلمات است و به جنبه‌های فرهنگی یا فنی (مانند راست‌چین‌سازی) توجه نمی‌کند.

- **جهانی‌سازی (Globalization - G11n):** یک فرآیند گسترده‌تر است که شامل طراحی و توسعه یک محصول به گونه‌ای است که بتواند به راحتی برای بازارهای مختلف (بین‌المللی) بومی‌سازی شود. جهانی‌سازی در مراحل اولیه طراحی محصول انجام می‌شود و هدف آن حذف هرگونه وابستگی به یک زبان یا منطقه خاص است. این فرآیند زیربنایی برای بومی‌سازی کارآمد فراهم می‌کند. برای مثال، استفاده از توابع زمانی استاندارد به جای توابع محلی در کدنویسی، یک جنبه از جهانی‌سازی است.
- **بومی‌سازی (Localization - L10n):** پس از جهانی‌سازی، بومی‌سازی مرحله‌ای است که در آن محصول با جزئیات فرهنگی، زبانی، و فنی یک منطقه‌ی خاص تطبیق داده می‌شود. این شامل:

- ترجمه متون: ترجمه دقیق و فرهنگی (با توجه به اصطلاحات محلی).
- راست‌چین‌سازی (RTL): تطبیق چیدمان عناصر، تراز متن، و جهت‌گیری کلی رابط کاربری برای زبان‌هایی مانند فارسی.
- قالب‌بندی تاریخ و زمان: استفاده از تقویم محلی (مانند شمسی)، فرمت‌های تاریخ و زمان مرسوم، و سیستم‌های اعداد.
- واحد پول: استفاده از واحد پول محلی و نمادهای آن.
- آدرس و شماره تلفن: تطبیق با فرمت‌های محلی آدرس و شماره تلفن.
- درگاه‌های پرداخت: یکپارچه‌سازی با سیستم‌های پرداخت بومی.
- تصاویر و نمادها: جایگزینی تصاویر یا نمادهایی که ممکن است در فرهنگ مقصد نامناسب یا بی‌معنا باشند.
- ملاحظات قانونی: رعایت قوانین و مقررات محلی (مانند حریم خصوصی داده‌ها).

۲-۳-۲) اهمیت بومی‌سازی در نرم‌افزار و وب

اهمیت بومی‌سازی در دنیای امروز، که رقابت جهانی فشرده‌ای وجود دارد و دسترسی به اطلاعات در سراسر جهان گسترده شده، بیش از پیش احساس می‌شود. دلایل اصلی اهمیت بومی‌سازی عبارت‌اند از:

- **افزایش دسترسی‌پذیری و پوشش بازار:** بومی‌سازی به شرکت‌ها امکان می‌دهد تا به بازارهای جدید و گسترده‌ای دسترسی پیدا کنند. محصولی که تنها به یک زبان ارائه می‌شود، فقط بخش کوچکی از مخاطبان جهانی را پوشش می‌دهد.

- **بهبود تجربه کاربری (UX):** کاربران ترجیح می‌دهند با محصولاتی کار کنند که به زبان مادری آنهاست و با فرهنگشان همخوانی دارد. یک محصول بومی‌سازی شده، حس آشنایی و راحتی بیشتری برای کاربر ایجاد می‌کند که منجر به افزایش رضایت و وفاداری می‌شود.
- **افزایش فروش و درآمد:** با دسترسی به بازارهای جدید و ارائه تجربه کاربری بهتر، احتمال افزایش فروش و درآمد به طور قابل توجهی بالا می‌رود.
- **کاهش هزینه‌های پشتیبانی:** وقتی محصول به طور کامل بومی‌سازی شده باشد، کاربران کمتر با مشکلات زبانی یا فرهنگی مواجه می‌شوند، در نتیجه نیاز به پشتیبانی فنی و زبانی کاهش می‌یابد.
- **اعتمادسازی و اعتبار برند:** محصولاتی که به طور کامل بومی‌سازی شده‌اند، نشان‌دهنده احترام به فرهنگ و نیازهای محلی هستند. این امر به اعتمادسازی و تقویت اعتبار برند در بازار مقصد کمک می‌کند.
- **رعایت الزامات قانونی و فرهنگی:** در برخی کشورها، بومی‌سازی برای رعایت قوانین و مقررات محلی (مانند قوانین مربوط به حریم خصوصی یا نمایش اطلاعات مالی) ضروری است.
- **مزیت رقابتی:** در بازارهایی که رقابت بالاست، محصول بومی‌سازی شده می‌تواند مزیت رقابتی قابل توجهی نسبت به رقبای خود که محصولاتشان بومی‌سازی نشده‌اند، ایجاد کند.

در مجموع، بومی‌سازی نه تنها یک سرمایه‌گذاری برای گسترش بازار است، بلکه یک ضرورت برای ایجاد ارتباط مؤثر با کاربران و موفقیت در محیط‌های بین‌المللی و به خصوص برای بازار وردپرس ایران محسوب می‌شود.

۴-۲) مروری بر مارکت پلیس‌های ایرانی (مثل ژاکت، راست‌چین)

مارکت پلیس‌ها یا بازارهای آنلاین، پلتفرم‌هایی هستند که امکان خرید و فروش محصولات یا خدمات را بین فروشندگان و خریداران فراهم می‌آورند. در اکوسیستم وردپرس ایران، چندین مارکت پلیس تخصصی وجود دارند که به طور ویژه بر عرضه قالب‌ها و افزونه‌های وردپرس، به خصوص نسخه‌های بومی‌سازی شده‌ی آنها، تمرکز دارند. این مارکت پلیس‌ها نقش حیاتی در تسهیل دسترسی کاربران ایرانی به محصولات وردپرسی با کیفیت و مطابق با نیازهای بومی ایفا می‌کنند. ژاکت و راست‌چین، دو نمونه برجسته از این مارکت پلیس‌ها هستند.

۲-۴-۱) معرفی مارکت پلیس های ایرانی فعال در حوزه وردپرس

• شرکت ژاکت (Zhaket)

ژاکت یکی از قدیمی ترین و بزرگترین مارکت پلیس های ایرانی در زمینه قالب ها و افزونه های وردپرس است. این پلتفرم از سال ها پیش فعالیت خود را آغاز کرده و توانسته است جامعه ی بزرگی از توسعه دهندگان و کاربران را به خود جذب کند.

ویژگی ها:

- ارائه طیف گسترده ای از قالب ها و افزونه های فارسی و راست چین شده.
- تضمین کیفیت و پشتیبانی محصولات توسط فروشندگان.
- سیستم امتیازدهی و نظرات کاربران برای ارزیابی محصولات.
- امکان ارائه محصولات توسط توسعه دهندگان مستقل.
- پشتیبانی از درگاه های پرداخت ایرانی.
- برگزاری دوره های آموزشی و رویدادهای مرتبط با وردپرس.
- نقش در بومی سازی: ژاکت با تایید و ارائه محصولات بومی سازی شده، به استانداردسازی و بهبود کیفیت این محصولات در بازار ایران کمک می کند. فروشندگان موظف به رعایت اصول خاصی در بومی سازی هستند تا محصولشان در این مارکت پلیس پذیرفته شود.



تصویر ۲-۳: شرکت ژاکت

• شرکت راست چین

راست چین نیز یکی دیگر از مارکت پلیس های محبوب و معتبر وردپرس در ایران است که به طور خاص بر محصولات فارسی و راست چین تمرکز دارد. این پلتفرم نیز با هدف فراهم آوردن محصولات با کیفیت برای کاربران فارسی زبان فعالیت می کند.

ویژگی‌ها:

- تمرکز ویژه بر محصولات فارسی و راست‌چین با کیفیت بالا.
- سیستم جامع بررسی و تایید محصولات قبل از انتشار.
- ارائه مستندات و راهنماهای کاربردی برای توسعه‌دهندگان و کاربران.
- پشتیبانی فعال از کاربران و فروشندگان.
- برگزاری کمپین‌های فروش و تخفیف‌های فصلی.
- نقش در بومی‌سازی: راست‌چین با الزامات سخت‌گیرانه‌تر در زمینه بومی‌سازی و تست محصولات، به عنوان یک مرجع برای کیفیت محصولات فارسی‌سازی شده عمل می‌کند. این پلتفرم تلاش می‌کند تا اطمینان حاصل کند که محصولات ارائه شده، کاملاً با نیازهای کاربران ایرانی سازگار هستند.



تصویر ۲-۴: شرکت راست چین

۲-۴-۲) بررسی ویژگی‌ها و خدمات آن‌ها

هر دو مارکت پلیس ژاکت و راست‌چین، خدمات مشابهی را با تفاوت‌هایی در رویکرد ارائه می‌دهند که به طور کلی شامل موارد زیر است:

- **کاتالوگ محصولات:** هر دو پلتفرم دارای کاتالوگ‌های گسترده‌ای از قالب‌ها و افزونه‌های وردپرس در دسته‌بندی‌های مختلف (فروشگاهی، شرکتی، آموزشی، شخصی و...) هستند.
- **پشتیبانی:** ارائه سیستم پشتیبانی از طریق تیکت، انجمن، یا چت آنلاین برای کمک به کاربران و فروشندگان.
- **سیستم پرداخت:** یکپارچگی با درگاه‌های پرداخت داخلی برای تسهیل فرآیند خرید برای کاربران ایرانی.
- **بخش فروشندگان:** امکان ثبت‌نام برای توسعه‌دهندگان به عنوان فروشنده و ارائه محصولات خود پس از گذراندن فرآیند تایید.

- **پیش‌نمایش (Demo) :** ارائه دمو یا پیش‌نمایش آنلاین برای قالب‌ها و افزونه‌ها تا کاربران بتوانند قبل از خرید، عملکرد و ظاهر محصول را مشاهده کنند.
- **لایسنس و به‌روزرسانی:** ارائه سیستم لایسنس برای محصولات و امکان دریافت به‌روزرسانی‌های آینده.
- **استانداردهای بومی‌سازی:** هر دو مارکت‌پلیس دارای دستورالعمل‌ها و استانداردهایی برای بومی‌سازی محصولات هستند که فروشندگان باید آن‌ها را رعایت کنند. این شامل الزام به راست‌چین‌سازی کامل، فارسی‌سازی صحیح، پشتیبانی از تقویم شمسی و هماهنگی با فونت‌های فارسی است.

این مارکت‌پلیس‌ها نقش مهمی در ایجاد یک بازار رقابتی و با کیفیت برای محصولات وردپرسی در ایران ایفا می‌کنند و به دلیل تحریم‌ها و عدم دسترسی آسان به مارکت‌های بین‌المللی، اهمیت آن‌ها دوچندان است. درک نحوه عملکرد و الزامات آن‌ها برای هر توسعه‌دهنده‌ای که قصد عرضه محصول بومی‌سازی شده را دارد، ضروری است.

۲-۵) مقالات یا پروژه‌های مشابه در این حوزه

هرچند موضوع بومی‌سازی محصولات نرم‌افزاری به طور کلی در ادبیات علمی و فنی مورد بحث قرار گرفته است، اما بومی‌سازی خاص قالب‌ها و افزونه‌های وردپرس برای مارکت‌پلیس‌های ایرانی، به دلیل ماهیت فنی و فرهنگی خاص خود، نیاز به بررسی عمیق‌تری دارد. در این بخش، به مرور برخی از مقالات، پایان‌نامه‌ها، و پروژه‌های عملی مشابه که به صورت مستقیم یا غیرمستقیم به این موضوع پرداخته‌اند، می‌پردازیم. این مرور به شناسایی نقاط قوت و ضعف تحقیقات پیشین و تعیین جایگاه تحقیق حاضر کمک می‌کند.

۲-۵-۱) بررسی مقالات و پایان‌نامه‌های مرتبط

- **تحقیقات در زمینه بومی‌سازی نرم‌افزار:** بسیاری از مقالات دانشگاهی به اصول و فرآیندهای کلی بومی‌سازی نرم‌افزار می‌پردازند. این مقالات معمولاً چارچوب‌های نظری، مدل‌های فرآیندی (مانند مدل GILT: Globalization, Internationalization, Localization, Translation) و ابزارهای عمومی بومی‌سازی را معرفی می‌کنند. برای مثال، تحقیقاتی که به بررسی چالش‌های بومی‌سازی رابط‌های کاربری گرافیکی (GUI) برای زبان‌های RTL می‌پردازند، می‌توانند نکات ارزشمندی در زمینه استایل‌دهی CSS و چیدمان عناصر ارائه دهند.

- **مطالعات موردی در بومی سازی CMS ها:** برخی پایان نامه ها یا پروژه ها ممکن است به بومی سازی CMS های خاصی (غیر از وردپرس) یا نسخه های قدیمی تر وردپرس پرداخته باشند. این مطالعات می توانند دیدگاه هایی در مورد چالش های عمومی و راه حل های فنی مربوط به CMS ها ارائه دهند، اما ممکن است به طور خاص به چالش های وردپرس یا مارکت پلیس های ایرانی نپردازند.
- **مقالات فنی در زمینه وردپرس:** تعداد زیادی مقاله فنی و بلاگ پست در زمینه توسعه قالب و افزونه وردپرس وجود دارد که به مباحثی مانند Internationalization (آماده سازی برای ترجمه)، استفاده از Text Domain و توابع ترجمه در وردپرس می پردازند. این منابع فنی، پایه های لازم برای انجام بومی سازی فنی را فراهم می کنند. هرچند این مقالات غالباً جنبه های عملیاتی را پوشش می دهند، اما کمتر به ابعاد جامع فرآیند بومی سازی و چالش های عرضه در مارکت پلیس ها می پردازند.
- **پروژه های عملی بومی سازی:** برخی از پروژه ها و مستندات آنلاین توسط توسعه دهندگان ایرانی، به صورت غیررسمی، به فرآیند فارسی سازی و راست چین سازی قالب ها و افزونه ها پرداخته اند. این منابع غالباً تجربیات شخصی و راهکارهای کدنویسی را به اشتراک می گذارند، اما معمولاً فاقد ساختار علمی و جامعیت یک تحقیق دانشگاهی هستند.

۲-۵-۲) تحلیل یافته های آن ها و شکاف تحقیقاتی

با وجود منابع فوق، یک شکاف تحقیقاتی مشخص در زمینه بومی سازی وردپرس برای مارکت پلیس های ایرانی وجود دارد. تحقیقات موجود غالباً یا به جنبه های بسیار کلی بومی سازی نرم افزار می پردازند، یا صرفاً بر روی نکات فنی خرد در توسعه وردپرس تمرکز دارند. کمتر تحقیقی به صورت جامع و سیستماتیک به بررسی تمامی ابعاد فرآیند بومی سازی یک قالب یا افزونه وردپرس (از انتخاب محصول تا عرضه آن در مارکت پلیس های ایرانی) پرداخته است. به ویژه، جنبه هایی مانند:

- **چالش های خاص فرهنگی و زبانی ایران:** مانند مسائل مربوط به تقویم شمسی، اعداد فارسی، و سازگاری با فونت های فارسی که فراتر از RTL صرف هستند.
- **مسائل مربوط به تحریم ها و لایسنس گذاری:** که دسترسی به منابع بین المللی را محدود می کند و نیاز به راه حل های بومی دارد.
- **الزامات خاص مارکت پلیس های ایرانی:** مانند فرآیندهای تایید، استانداردهای محتوایی، و نحوه پشتیبانی که در مارکت های بین المللی متفاوت است.

- **مطالعه موردی جامع و عملی:** یک مثال عملی کامل که تمامی مراحل بومی‌سازی را با جزئیات فنی و چالش‌های آن مستند کند، کمتر در منابع دانشگاهی موجود است.

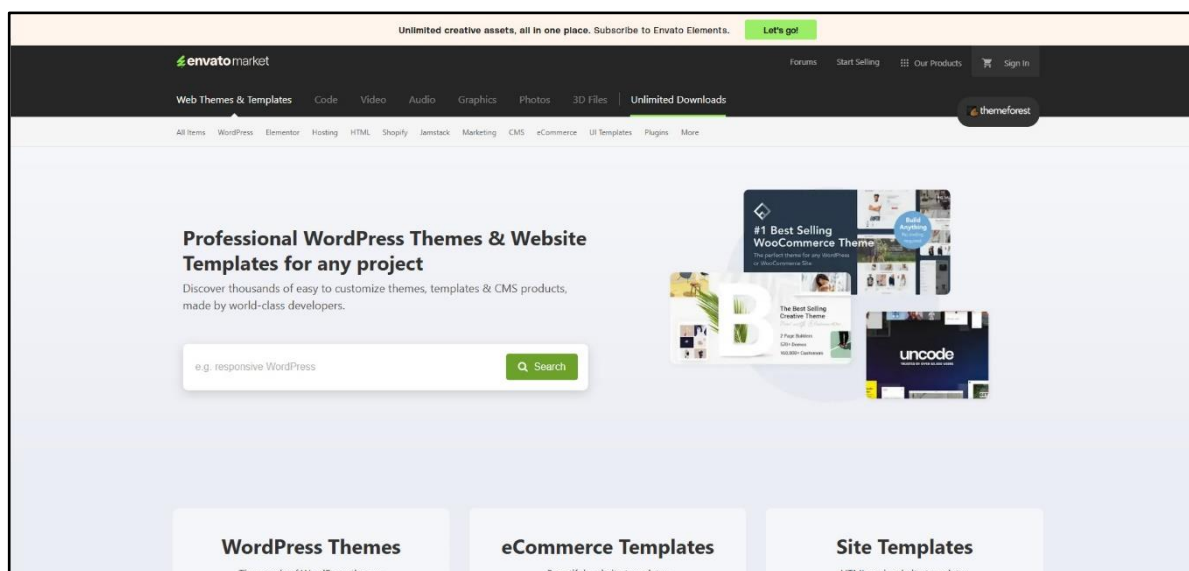
این پروژه با هدف پر کردن این شکاف تحقیقاتی و ارائه یک راهنمای جامع و عملی، تمامی ابعاد فرآیند بومی‌سازی قالب‌ها و افزونه‌های وردپرس برای مارکت‌پلیس‌های ایرانی را، از مراحل فنی تا نکات مربوط به عرضه، بررسی خواهد کرد. مطالعه موردی عملی در فصل سوم، به عنوان یک بخش کلیدی، تجربیات و راهکارهای عملی را در اختیار خواننده قرار خواهد داد.

فصل ۳: روش اجرای تحقیق و تجزیه و تحلیل داده‌ها (یافته‌ها)

در این فصل، به تشریح روش‌شناسی، مراحل اجرایی و یافته‌های حاصل از بومی‌سازی عملی یک قالب بین‌المللی وردپرس جهت عرضه‌ی آن در مارکت‌پلیس‌های ایرانی پرداخته می‌شود. این بخش، در واقع پیاده‌سازی عملی چارچوب نظری مطرح‌شده در فصول پیشین و پاسخ به بخش‌های کاربردی سؤالات تحقیق است. تمامی مراحل، ابزارهای به‌کارگرفته‌شده و چالش‌های مواجهه‌شده به صورت گزارشی و تحلیلی ارائه می‌گردد تا خواننده با رویکردهای عملی و راهکارهای فنی مورد استفاده در فرآیند بومی‌سازی آشنا شود.

۳-۱) انتخاب یک قالب از مارکت بین‌المللی (مطالعه موردی: ThemeForest)

انتخاب قالب مناسب برای بومی‌سازی، گام نخست در این مطالعه‌ی موردی بود. معیارهای اصلی انتخاب، شامل محبوبیت قالب در بازار جهانی، پشتیبانی فعال توسط توسعه‌دهنده، و وجود قابلیت‌های پیچیده (مانند سیستم رزرواسیون یا فیلترهای پیشرفته) بود که امکان بررسی جامع چالش‌های بومی‌سازی را فراهم آورد. با توجه به این معیارها، قالب تراولر (Traveler) از وب‌سایت ThemeForest انتخاب شد [۱]. این قالب، به دلیل ماهیت گردشگری و قابلیت‌های گسترده‌ای نظیر سیستم رزرواسیون هتل و پرواز، گزینه‌ای مناسب برای بررسی ابعاد مختلف بومی‌سازی، از جمله شمس‌سازی تاریخ‌ها و یکپارچه‌سازی درگاه پرداخت، محسوب می‌شد. فرآیند خرید قالب، شامل رعایت نکات مربوط به لایسنسینگ و دانلود فایل‌های اصلی محصول از وب‌سایت ThemeForest بود.



تصویر ۳-۱: سایت ThemeForest

۳-۱-۱) معرفی ThemeForest و معیارهای انتخاب قالب

ThemeForest به عنوان یکی از بزرگ‌ترین و شناخته‌شده‌ترین بازارهای آنلاین جهانی برای قالب‌ها و پوسته‌های وب‌سایت، زیرمجموعه‌ی شرکت Envato Market است. این پلتفرم، میزبان هزاران قالب برای سیستم‌های

مدیریت محتوای مختلف از جمله وردپرس است که توسط توسعه‌دهندگان مستقل طراحی و عرضه می‌شوند و معمولاً دارای پشتیبانی و به‌روزرسانی‌های منظم هستند.

در فرآیند انتخاب قالب برای این تحقیق، معیارهای مشخصی مد نظر قرار گرفت:

- **قابلیت‌های جستجو و فیلتر:** بررسی امکان استفاده از فیلترهای پیشرفته در ThemeForest برای محدود کردن گزینه‌ها بر اساس دسته‌بندی‌ها (مانند فروشگاه، شرکتی، وبلاگی)، ویژگی‌ها (مانند واکنش‌گرا بودن، سازگاری با صفحه‌سازها) و کلمات کلیدی مرتبط.
- **بررسی پیش‌نمایش (Demo) زنده:** هر قالب دارای پیش‌نمایش زنده‌ای است که امکان ارزیابی بصری و عملکردی قالب را قبل از خرید فراهم می‌آورد. این بررسی به شناسایی پیچیدگی‌های احتمالی در رابط کاربری کمک می‌کند.
- **مطالعه‌ی جامع توضیحات و ویژگی‌ها:** صفحه‌ی محصول هر قالب حاوی توضیحات جامع در مورد ویژگی‌ها، قابلیت‌ها، افزونه‌های مورد نیاز، مستندات و نیازمندی‌های سیستمی است که برای برنامه‌ریزی بومی‌سازی مورد استفاده قرار گرفت.
- **تحلیل نظرات و امتیازات کاربران:** نظرات و امتیازات سایر خریداران به عنوان یک معیار مهم برای ارزیابی کیفیت، پایداری و سطح پشتیبانی قالب مورد توجه قرار گرفت.
- **توجه به تاریخ به‌روزرسانی و حجم فروش:** قالب‌هایی که به طور منظم به‌روزرسانی می‌شوند و تعداد فروش بالایی دارند، معمولاً نشان‌دهنده‌ی پشتیبانی فعال و کیفیت قابل‌قبول هستند که ریسک‌های فرآیند بومی‌سازی را کاهش می‌دهند.

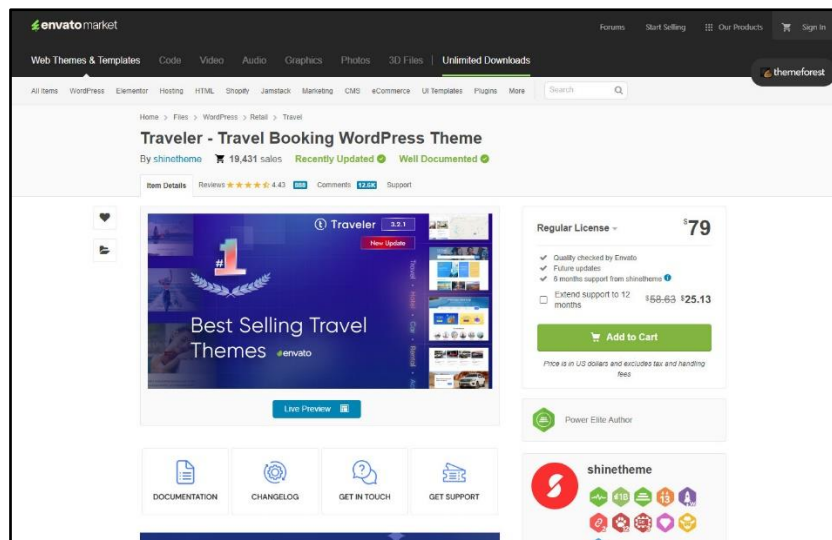
۳-۱-۲) ملاحظات مهم در انتخاب قالب با هدف بومی‌سازی

انتخاب قالب با هدف بومی‌سازی برای مارکت‌پلیس‌های ایرانی، نیازمند دقت به نکاتی فراتر از یک خرید عادی بود. این نکات به طور مستقیم بر سهولت و موفقیت فرآیند بومی‌سازی تأثیرگذار هستند:

- **پشتیبانی از Internationalization (i18n):** یکی از مهم‌ترین فاکتورها، کدنویسی قالب به گونه‌ای بود که از فرآیند بین‌المللی‌سازی پشتیبانی کند. این بدان معناست که تمامی رشته‌های متنی (Strings) درون قالب باید قابل ترجمه باشند.
- **وجود فایل‌های POT:** حضور فایل POT (Portable Object Template) در ساختار قالب، به عنوان لیستی از تمامی رشته‌های متنی قابل ترجمه، نقطه‌ی شروعی حیاتی برای فرآیند ترجمه بود.

- **کیفیت کدنویسی و مستندات:** قالب با کدنویسی تمیز، استاندارد و مستندات جامع، بومی سازی را آسان تر می کرد. دسترسی به مستندات دقیق در مورد ساختار فایل ها، توابع و نحوه ی سفارشی سازی، در حل مشکلات احتمالی کمک کننده بود.
- **پشتیبانی از RTL (Right-to-Left):** اگرچه هدف اصلی این تحقیق راست چین سازی قالب بود، اما وجود یک فایل CSS اختصاصی برای RTL (مانند rtl.css) یا پشتیبانی پیش فرض از RTL، می توانست فرآیند را تسریع کند. حتی در صورت عدم پشتیبانی کامل، وجود کلاس های CSS مناسب و ساختار HTML منطقی، کار را آسان تر می ساخت.
- **عدم وابستگی به فونت های خاص انگلیسی:** قالب هایی که وابستگی شدیدی به فونت های انگلیسی داشتند، ممکن بود در جایگزینی با فونت های فارسی مناسب، چالش برانگیز باشند. قابلیت تغییر فونت از طریق پنل تنظیمات قالب، یک مزیت محسوب می شد.
- **ملاحظات افزونه ها:** سازگاری قالب با افزونه های کلیدی و قابلیت بومی سازی این افزونه ها نیز مورد بررسی قرار گرفت. به عنوان مثال، در صورت استفاده از WooCommerce برای قابلیت های فروشگاه، بومی سازی افزونه های مرتبط با آن نیز اهمیت داشت.
- **سازگاری با نسخه ی PHP و وردپرس:** اطمینان از سازگاری قالب با آخرین نسخه های PHP و وردپرس برای جلوگیری از مشکلات امنیتی و عملکردی، ضروری بود.

در این مطالعه ی موردی، **قالب تراولر (Traveler)** از ThemeForest برای بومی سازی انتخاب شد. این قالب، به عنوان یک قالب جامع برای وبسایت های رزرو و گردشگری با قابلیت های پیچیده، انتخاب گردید تا چالش های متعددی را در فرآیند بومی سازی ایجاد کند و منجر به تجربه ی عملی عمیقی در این تحقیق شود.



تصویر ۳-۲: قالب Traveler

۳-۲) راه اندازی محیط توسعه‌ی محلی و نصب وردپرس

برای انجام فرآیند بومی‌سازی و توسعه‌ی قالب وردپرس به صورت کارآمد و بدون تأثیر بر محیط آنلاین، راه اندازی یک محیط توسعه‌ی محلی (Local Development Environment) ضروری بود. این محیط، امکان اجرای و تست وبسایت وردپرسی را بر روی رایانه‌ی شخصی و بدون نیاز به سرور آنلاین و هزینه‌های مربوط به آن فراهم می‌آورد.

۳-۲-۱) اهمیت محیط توسعه‌ی محلی

استفاده از یک محیط توسعه‌ی محلی دارای مزایای قابل توجهی بود:

- **ایمنی و پایداری:** تمامی تغییرات و آزمایش‌ها در محیطی ایزوله انجام شد که هیچ تأثیری بر وبسایت‌های آنلاین یا اطلاعات واقعی نداشت. این امر خطر از دست دادن اطلاعات یا خراب شدن وبسایت اصلی را از بین می‌برد.
- **سرعت بالا:** عملکرد وبسایت در محیط محلی به دلیل عدم وابستگی به سرعت اینترنت و سرورهای خارجی، بسیار سریع‌تر بود که منجر به افزایش بهره‌وری در فرآیند توسعه و اشکال‌زدایی شد.
- **دسترسی آفلاین:** امکان کار بر روی وبسایت در هر زمان و مکانی، حتی بدون اتصال به اینترنت، فراهم بود.
- **کنترل کامل:** دسترسی کامل به تمامی فایل‌ها، پایگاه داده و تنظیمات سرور محلی، امکان اعمال تغییرات عمیق‌تر و تست‌های جامع‌تر را فراهم می‌آورد.

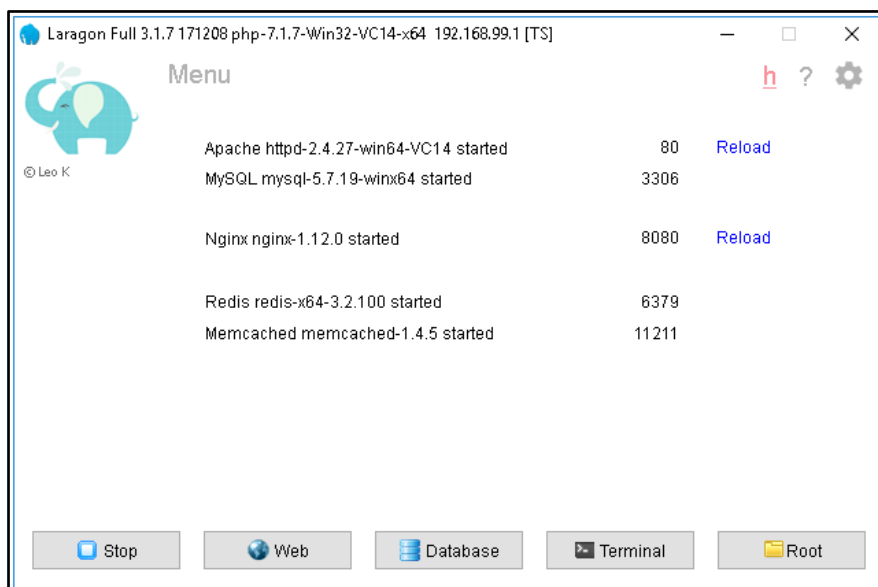
- تست و اشکال زدایی آسان: خطاهای کدنویسی یا مشکلات عملکردی به سرعت شناسایی و رفع می‌شدند، زیرا تمامی گزارش‌ها و لاگ‌ها در دسترس بودند.

۳-۲-۲) نصب و راه‌اندازی Laragon

لارگون (Laragon) به عنوان یک محیط توسعه‌ی محلی جامع و قدرتمند برای PHP، Node.js، Python و Ruby انتخاب شد که به طور خاص برای توسعه‌دهندگان وب طراحی شده است. این نرم‌افزار، سرور آپاچی (Apache)، پایگاه داده MySQL (یا MariaDB) و مفسر PHP را به صورت پیش‌فرض شامل می‌شود و فرآیند نصب وردپرس و سایر فریم‌ورک‌ها را ساده می‌کند.

مراحل نصب و راه‌اندازی Laragon به شرح زیر انجام شد:

۱. **دانلود و نصب Laragon:** جدیدترین نسخه‌ی Laragon (نسخه‌ی کامل) از وبسایت رسمی آن (laragon.org) دانلود و بر روی سیستم عامل نصب گردید.
۲. **پیکربندی اولیه و راه‌اندازی سرور:** پس از نصب، Laragon اجرا و دکمه‌ی "Start All" برای راه‌اندازی سرویس‌های Apache و MySQL کلیک شد.
۳. **ایجاد دامنه‌ی مجازی (Virtual Host) خودکار Laragon:** به صورت خودکار برای پروژه‌های جدید، دامنه‌های مجازی ایجاد می‌کرد (مانند `myproject.test`) که امکان دسترسی به پروژه‌های مختلف را به صورت هم‌زمان فراهم می‌آورد.



تصویر ۳-۳: محیط Laragon

۳-۲-۳) نصب داپلیکیتور

داپلیکیتور (Duplicator) یک افزونه‌ی وردپرس است که برای انتقال، کپی‌برداری و پشتیبان‌گیری از وبسایت‌های وردپرسی مورد استفاده قرار می‌گیرد. در این تحقیق، از داپلیکیتور به منظور شبیه‌سازی فرآیند دریافت یک وبسایت وردپرسی موجود (که قالب در آن نصب شده) و آماده‌سازی آن برای بومی‌سازی و توسعه در محیط محلی استفاده شد.

مراحل نصب و استفاده از داپلیکیتور به شرح زیر بود:

۱. **نصب و فعال‌سازی افزونه:** داپلیکیتور از طریق پنل مدیریت وردپرس نصب و فعال‌سازی شد.
۲. **ایجاد بسته‌ی داپلیکیتور:** پس از فعال‌سازی، یک بسته‌ی کامل از وبسایت (شامل فایل‌ها و پایگاه داده) با استفاده از گزینه‌ی "ایجاد جدید" در بخش "بسته‌ها"ی داپلیکیتور ساخته شد. این فرآیند منجر به تولید یک فایل آرشیو (.zip یا .daf) و یک فایل نصب‌کننده (installer.php) گردید.
۳. **انتقال و استخراج در محیط جدید:** فایل‌های آرشیو و نصب‌کننده به پوشه‌ی ریشه‌ی وبسایت جدید در محیط Laragon منتقل شدند. سپس، با مراجعه به آدرس installer.php در مرورگر، مراحل استخراج و بازسازی وبسایت در محیط محلی دنبال و تکمیل شد.

Duplicator - Installer

1 Deploy 2 Update 3 Test

version: 1.1.24 [help]

Step 1: Deploy Files & Database

Requirements: Pass

MySQL Database

Action: ☐ Create New Database ☒ Connect and Remove All Data

Host: localhost Port: 3306

Name: new or existing database name

User: valid database username

Password: valid database user password

Test Connection

Need Setup Help...

Advanced Options...

*Create the database and users from the installer with Duplicator Pro! - Requires cPanel.

WARNINGS & NOTICES

Disclaimer: This plugin require above average technical knowledge. Please use it at your own risk and always back up your database and files beforehand using another backup system besides the Duplicator. If you're not sure about how to use this tool then please enlist the guidance of a technical professional. Always test this installer in a sandbox environment before trying to deploy into a production setting.

☒ I have read all warnings & notices

The remove action will delete all tables and data from the database!

Run Deployment

تصویر ۳-۴: نصب Duplicator

۳-۲-۴) نصب وردپرس در محیط محلی (برای شروع از صفر)

در صورتی که هدف، شروع یک پروژه وردپرسی از صفر باشد (که در این تحقیق، این مرحله برای نصب قالب بومی سازی شده در یک محیط پاک مورد استفاده قرار می‌گرفت)، وردپرس به صورت دستی در محیط Laragon نصب می‌شد:

۱. **دانلود و استخراج فایل‌های وردپرس:** آخرین نسخه وردپرس از وبسایت رسمی آن دانلود و در یک پوشه جدید در مسیر پروژه‌های Laragon استخراج شد.
۲. **ایجاد پایگاه داده:** یک پایگاه داده جدید (MySQL) از طریق ابزارهای مدیریت پایگاه داده Laragon (مانند phpMyAdmin) ایجاد گردید.
۳. **تکمیل نصب از طریق مرورگر:** با مراجعه به آدرس دامنه مجازی پروژه جدید در مرورگر، مراحل نصب وردپرس شامل وارد کردن اطلاعات پایگاه داده و تنظیم اطلاعات مدیر وبسایت تکمیل شد.

۳-۳) بررسی ساختار فایل‌های قالب، ترجمه متون، فونت و RTL

پس از راه‌اندازی محیط محلی و نصب قالب (تراولر)، گام بعدی بررسی عمیق ساختار فایل‌های قالب و شناسایی نقاطی بود که نیاز به بومی‌سازی داشتند. این مرحله، پایه و اساس تمامی تغییرات بعدی را تشکیل می‌داد.

۳-۳-۱) ساختار فایل‌های قالب انتخابی (تراولر)

همانطور که در فصل دوم اشاره شد، قالب‌های وردپرس از ساختار فایل‌های خاصی پیروی می‌کنند. قالب تراولر نیز به عنوان یک قالب پیچیده و چندمنظوره، دارای ساختاری غنی از فایل‌ها و پوشه‌ها بود. بررسی این ساختار شامل موارد زیر بود:

- **فایل‌های اصلی PHP:** شناسایی فایل‌هایی مانند `index.php`، `header.php`، `footer.php` و `functions.php` و فایل‌های مربوط به صفحات خاص (مانند `single.php`، `page.php` و `archive.php`) برای درک جریان کد و نقاط تعامل.
- **پوشه‌ی `inc` یا `includes`:** این پوشه معمولاً شامل فایل‌های کمکی، توابع اختصاصی قالب، کلاس‌ها و تعریف پست‌تایپ‌ها و تگس‌نوم‌های سفارشی بود. بسیاری از رشته‌های متنی و منطق‌های برنامه‌نویسی در این فایل‌ها قرار داشتند.

- پوشه‌ی **assets** (شامل **css, js, images**): بررسی فایل‌های استایل‌دهی CSS (مانند **style.css** و سایر فایل‌های CSS مرتبط)، فایل‌های جاوااسکریپت (مانند **main.js** یا **custom.js**) و پوشه‌ی تصاویر انجام شد. تغییرات مربوط به راست‌چین‌سازی و فونت‌ها عمدتاً در فایل‌های CSS اعمال می‌شدند.
- پوشه‌ی **languages**: این پوشه به عنوان محلی برای نگهداری فایل‌های ترجمه (با پسوند **.po** و **.mo**) شناسایی شد. وجود این پوشه و فایل‌های **POT** نشان‌دهنده‌ی آمادگی قالب برای ترجمه بود.
- پوشه‌ی **template-part**: در قالب‌های مدرن، بخش‌های مختلف قالب به صورت ماژولار در این پوشه قرار می‌گرفتند که نشانه‌ای از ساختار کد تمیزتر و قابل استفاده‌تر بود.

۳-۳-۲) دانش پایه HTML و CSS مورد نیاز برای بومی‌سازی

اگرچه برای بومی‌سازی کامل، نیاز به دانش عمیق برنامه‌نویسی نیست، اما آشنایی با مبانی HTML و CSS برای انجام تغییرات بصری و راست‌چین‌سازی ضروری بود.

- **HTML (HyperText Markup Language)**: به عنوان زبان ساختاردهی محتوای وب، درک تگ‌های HTML (مانند **<a>**, **<h1>**, **<p>**, **<div>**) و ویژگی‌های آن‌ها (مانند **class**, **id**, **dir="rtl"**) برای شناسایی و افزودن ویژگی‌های RTL به عناصر مختلف قالب اهمیت داشت.
- **CSS (Cascading Style Sheets)**: به عنوان زبان استایل‌دهی به محتوای HTML، تمامی تغییرات ظاهری مانند فونت، رنگ، اندازه، فاصله و جهت‌گیری عناصر با CSS انجام شد. برای راست‌چین‌سازی، تغییراتی نظیر:

```
direction: rtl;
text-align: right;
float: right;
```

و معکوس کردن مقادیر **margin** و **padding** برای عناصر خاص ضروری بود.

۳-۳-۳) رویکرد راست‌چین‌سازی قالب استاتیک

راست‌چین‌سازی (RTL) به عنوان یکی از مهم‌ترین جنبه‌های بومی‌سازی برای زبان فارسی، شامل تغییر جهت نمایش متن و عناصر از چپ به راست (LTR) به راست به چپ (RTL) بود. در این مطالعه، در صورتی که قالب دارای یک فایل **rtl.css** از پیش تعریف‌شده نبود، یا این فایل نیاز به تکمیل داشت، تغییرات به صورت دستی یا با کمک ابزارها اعمال شدند.

مراحل اصلی راست‌چین‌سازی شامل:

- تعریف "dir="rtl" در تگ <html>: این ویژگی به مرورگر اعلام می‌کند که جهت کلی سند از راست به چپ است.
- استفاده از direction: rtl در CSS: این ویژگی برای <body> یا عناصر خاصی که نیاز به راست‌چین شدن داشتند، اعمال شد.
- تنظیم تراز متن (text-align): بسیاری از متون به صورت راست تراز شدند (text-align: right).
- تغییر فلوتینگ (float): عناصری که در حالت LTR به چپ فلوت شده بودند (مانند سایدبارها)، به راست فلوت شدند (float: right).
- معکوس کردن حاشیه‌ها و فاصله‌ها (Margins & Paddings): مقادیر margin-left به margin-right و برعکس تبدیل شدند. همین منطق برای padding نیز صدق می‌کرد.
- تنظیمات مربوط به پس‌زمینه (Background Position): در صورت نیاز، تصاویر پس‌زمینه‌ای که برای چپ یا راست تنظیم شده بودند، معکوس شدند.
- سازگاری با فرم‌های ورودی: فیلدهای فرم، دکمه‌ها و سایر عناصر ورودی نیز به درستی راست‌چین شدند.
- استفاده از ابزارهای کمکی: در صورت امکان، از ابزارها یا فریم‌ورک‌های CSS مانند Bootstrap RTL برای تسریع این فرآیند کمک گرفته شد.

```
1 @media (min-width: 768px) {
2   .text-md-start {
3     margin-right: auto !important;
4   }
5 }
6
7 img[src="assets/img/undraw_svg_2.svg"] {
8   transform: scaleX(-1);
9 }
10
11 img[src="assets/img/undraw_svg_3.svg"] {
12   transform: scaleX(-1);
13 }
14
15 .section-heading {
16   background: linear-gradient(-45deg, #274685, #3db3c5);
17   -webkit-background-clip: text !important;
18   -webkit-text-fill-color: transparent !important;
19 }
20
21 @media (min-width: 768px) {
22   .text-md-start {
23     text-align: right !important;
24   }
25 }
26
27 /* MIN WIDTH 992px */
28 @media (min-width: 992px) {
29   .text-lg-start {
30     text-align: right !important;
31   }
32 }
```

تصویر ۳-۵: فایل rtl-style.css

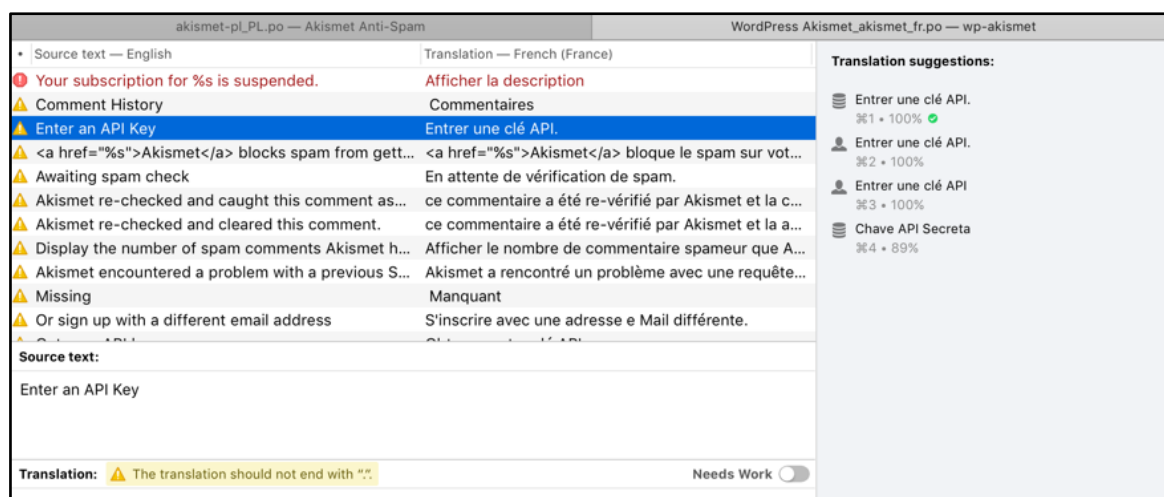
۳-۴) ابزارهای مورد استفاده (Poedit، Loco Translate، ویرایشگر کد)

برای انجام فرآیند بومی سازی به صورت مؤثر و کارآمد، استفاده از ابزارهای تخصصی ضروری بود. این ابزارها به ترجمه متون، مدیریت فایل های زبان و اشکال زدایی کمک می کنند.

۳-۴-۱) معرفی و کاربرد ابزارهای ترجمه (Poedit، Loco Translate)

• Poedit:

- **معرفی Poedit:** به عنوان یک نرم افزار ویرایشگر رایگان و متن باز برای فایل های gettext (با پسوندهای .po و .mp) مورد استفاده قرار گرفت. این ابزار به طور گسترده ای برای ترجمه ی نرم افزارها، وبسایت ها و قالب ها/افزونه های وردپرس استفاده می شود.
- **کاربرد در بومی سازی:** در این تحقیق، Poedit برای باز کردن فایل POT، قالب، ترجمه ی رشته های متنی انگلیسی به فارسی، و سپس ذخیره ی خودکار فایل .mo (نسخه ی کامپایل شده و قابل استفاده توسط وردپرس) در کنار فایل .po مورد استفاده قرار گرفت. همچنین، برای به روزرسانی ترجمه ها و اضافه کردن رشته های جدید، قابلیت مقایسه ی فایل POT، به روزرسانی شده با فایل .po موجود در Poedit کاربردی بود.



تصویر ۳-۶: تصویر محیط برنامه Poedit

• Loco Translate:

- **معرفی Loco Translate:** به عنوان یک افزونه وردپرس، امکان ترجمه فایل‌ها و افزونه‌ها را به صورت مستقیم از طریق پنل مدیریت وردپرس فراهم می‌کند. این افزونه به خصوص برای کاربرانی که دانش فنی کمتری دارند، بسیار کاربردی است.
- **کاربرد در بومی‌سازی Loco Translate:** امکان ترجمه فایل‌های ترجمه را از طریق داشبورد وردپرس فراهم کرد. این افزونه با فرمت‌های po و mo کار می‌کند و قابلیت اسکن خودکار قالب یا افزونه برای یافتن رشته‌های جدید قابل ترجمه را ارائه می‌دهد که در مدیریت آسان فایل‌های زبان مؤثر بود.



تصویر ۳-۷: افزونه وردپرس Loco Translate

۳-۴-۲) ویرایشگرهای کد (مثل VS Code) و اهمیت آن‌ها

ویرایشگر کد به عنوان ابزاری اساسی در فرآیند بومی‌سازی به کار گرفته شد. از طریق این ابزار، دسترسی به فایل‌های قالب و افزونه، اعمال تغییرات لازم در کد PHP، CSS، JavaScript و رفع مشکلات احتمالی امکان‌پذیر بود.

- **VS Code (Visual Studio Code):** این ویرایشگر کد رایگان، سبک و بسیار قدرتمند توسط مایکروسافت توسعه یافته است و به عنوان ویرایشگر اصلی در این مطالعه مورد استفاده قرار گرفت.
- **اهمیت در بومی‌سازی VS Code:** با قابلیت‌هایی نظیر Syntax Highlighting (رنگی کردن کدها برای خوانایی بهتر)، IntelliSense (تکمیل خودکار کد)، Debugger (امکان اشکال‌زدایی مستقیم) و پشتیبانی از Extensions (افزونه‌های متعدد برای زبان‌های برنامه‌نویسی و ابزارهای توسعه)، فرآیند ویرایش فایل‌های CSS برای راست‌چین‌سازی و فونت، اصلاح فایل‌های PHP و JavaScript برای

تطبیق با تقویم شمسی و درگاه‌های پرداخت، و جستجو و جایگزینی سریع رشته‌های متنی یا کدهای خاص را به شدت تسهیل کرد.

۳-۴-۳) کار با Composer و افزودن DD به پروژه برای دیباگ

Composer:

- **معرفی Composer:** به عنوان یک ابزار مدیریت وابستگی‌ها (Dependency Manager) برای PHP، امکان نصب، به‌روزرسانی و مدیریت آسان کتابخانه‌ها و بسته‌های مورد نیاز پروژه‌ی PHP را فراهم می‌کند.
- **کاربرد در بومی‌سازی:** در پروژه‌ی بومی‌سازی قالب تراولر، Composer برای افزودن کتابخانه‌های PHP خاصی برای کار با تقویم شمسی (مانند jdatetime) مورد استفاده قرار گرفت. به عنوان مثال، دستور `composer require miladrahimi/jdatetime` برای نصب کتابخانه‌ی `jdatetime` در پوشه‌ی `vendor` پروژه و ایجاد یا به‌روزرسانی فایل `autoload.php` اجرا شد تا امکان استفاده از کلاس‌های آن به راحتی فراهم آید.

DD (Dump and Die):

- **معرفی dd():** یک تابع کمکی در بسیاری از فریم‌ورک‌های PHP (مانند Laravel) است که برای اشکال‌زدایی (Debugging) استفاده می‌شود. این تابع، مقادیر یک یا چند متغیر را نمایش داده و سپس اجرای اسکریپت را متوقف می‌کند.
- **کاربرد در بومی‌سازی:** در هنگام کار با منطق‌های پیچیده‌ی قالب یا افزونه (مثلاً در پردازش تاریخ‌ها یا تراکنش‌های پرداخت)، `dd()` ابزار بسیار مفیدی برای ردیابی مشکلات و اطمینان از صحت مقادیر بود. برای استفاده از `dd()` در پروژه‌ی وردپرس (که به طور پیش‌فرض آن را ندارد)، این تابع از طریق Composer و با نصب پکیج‌هایی مانند `symfony/var-dumper` یا `spatie/macroable` و سپس تعریف یک تابع کمکی ساده، به پروژه اضافه شد.

مثال کد ساده برای کاربرد `dd()`:

```
PHP
<?php
$data = [
    'name' => 'John Doe',
```

```
'email' => 'john.doe@example.com'  
];  
dd($data); // This will output the content of $data and stop execution.  
echo 'This line will not be executed.';  
?>
```

۳-۵) مستندسازی فرآیند فارسی سازی قالب (مطالعه موردی: قالب تراولر)

این بخش به صورت جامع به مراحل عملی فارسی سازی قالب تراولر، با تمرکز بر چالش ها و راهکارهای فنی مورد استفاده در این مطالعه موردی می پردازد. این مراحل، نشان دهنده اقدامات عملی انجام شده در این تحقیق هستند.

۳-۵-۱) نصب قالب تراولر و افزونه های مورد نیاز

پس از راه اندازی محیط محلی با Laragon، گام بعدی نصب قالب تراولر در وردپرس بود. این فرآیند شامل مراحل استاندارد نصب قالب و سپس نصب افزونه های همراه (Required Plugins) است که قالب برای عملکرد صحیح به آنها نیاز دارد.

- **نصب قالب:** فایل zip قالب تراولر که از ThemeForest خریداری شده بود، به بخش "نمایش" (Appearance) و سپس "قالب ها" (Themes) در پنل مدیریت وردپرس آپلود و فعال شد. در این مطالعه، پس از نصب قالب اصلی (Parent Theme)، یک Child Theme برای قالب تراولر ایجاد و فعال شد تا تغییرات سفارشی در به روزرسانی های آینده قالب اصلی از بین نروند.
- **نصب افزونه های مورد نیاز:** پس از فعال سازی قالب، پیغامی در پنل مدیریت وردپرس برای نصب افزونه های مورد نیاز (Required Plugins) و افزونه های توصیه شده (Recommended Plugins) ظاهر شد. این افزونه ها شامل افزونه های اصلی قالب برای رزرو، مدیریت مشتریان، و یا سایر قابلیت های خاص قالب تراولر بودند که تمامی آنها از طریق گزینه ی "Begin installing plugins" نصب و فعال شدند. لازم به ذکر است که بسیاری از این افزونه ها نیز به بومی سازی جداگانه نیاز داشتند، زیرا بخشی از محتوای قابل ترجمه و عملکرد قالب را تشکیل می دادند.

۳-۵-۲) رفع تحریم لایسنس بین الملل

یکی از چالش های مهم برای توسعه دهندگان و کاربران در ایران، مسائل مربوط به تحریم ها و محدودیت های دسترسی به لایسنس ها و سرویس های بین المللی است. بسیاری از قالب ها و افزونه های پریمیوم نیاز به لایسنس برای دریافت به روزرسانی ها و پشتیبانی دارند که این لایسنس ها از طریق API های بین المللی بررسی می شوند.

راهکارهای عمومی برای رفع این مشکل عبارت‌اند از:

- استفاده از نسخه‌های GPL: برخی قالب‌ها و افزونه‌ها دارای نسخه‌های General Public License هستند که امکان استفاده بدون محدودیت لایسنس را فراهم می‌کنند، اما ممکن است فاقد پشتیبانی مستقیم یا به‌روزرسانی‌های خودکار باشند.
- پشتیبانی محلی توسط مارکت پلیس: در ایران، مارکت پلیس‌هایی مانند ژاکت و راست‌چین، خود لایسنس محصولات بین‌المللی را خریداری کرده و به‌روزرسانی‌ها را به صورت محلی در اختیار خریداران قرار می‌دهند. این رویکرد به کاربران ایرانی امکان می‌دهد تا بدون نگرانی از تحریم‌ها به نسخه‌های قانونی و به‌روز دسترسی داشته باشند.
- بلاک کردن API لایسنس: در برخی موارد، برای جلوگیری از مشکلاتی که بررسی لایسنس بین‌المللی ممکن است ایجاد کند (مانند درخواست‌های مکرر به سرورهای خارجی که منجر به کندی یا خطا می‌شود)، API‌های مربوط به بررسی لایسنس از طریق فایل hosts یا فایروال بلاک می‌شوند. این روش تنها در محیط توسعه‌ی محلی توصیه می‌شود و برای وبسایت‌های آنلاین مشکل‌ساز خواهد بود.

در این مطالعه‌ی موردی، از آنجایی که قالب از طریق یک مارکت پلیس داخلی (که خود لایسنس را مدیریت می‌کند) تهیه شده بود، چالش تحریم لایسنس به صورت مستقیم مواجه نشد. با این حال، شناخت این چالش‌ها برای توسعه‌دهندگان مستقل ضروری است.

```
1 <?php
2 // $purchase_code = get_option('envato_purchase_code');
3 // $has_error = STInput::get('has_error', '');
4 $purchase_code = 1;
5 $has_error = null;
6 $has_domain = STInput::get('domain', '');
7 $deregister = STInput::get('deregister', '');
8 $has_register = STAdmin\landingpage::checkValidatePurchaseCode($purchase_code);
9
10 if (empty($has_error)) {
11     if (empty($purchase_code)) {
12         if ($has_register) {
13             ?>
14             <div class="updated" style="padding: 8px 10px 5px 10px !important; margin-left: 0px; margin-top: 15px;">
15                 <?php _e('Thank you for registration our theme.', 'traveler'); ?>
16             </div>
17             <?php
18         } else {
19             ?>
20             <div class="error" style="padding: 8px 10px 5px 10px !important; margin-left: 0px; margin-top: 15px;">
21                 <?php _e('Your Purchase Code is invalid', 'traveler'); ?>
22             </div>
23             <?php
24         }
25     }
26 } else {
27     if (in_array($has_error, array('1', '2', '3', '5'))) {
28         ?>
29         <div class="error" style="padding: 8px 10px 5px 10px !important; margin-left: 0px; margin-top: 15px;">
30             <?php
31             if ($has_error == '1') {
32                 _e('Your Purchase Code is empty', 'traveler');
33             } elseif ($has_error == '2') {
34                 _e('Your Purchase Code is invalid', 'traveler');
35             } elseif ($has_error == '3') {
36                 // Handle error case 3
37                 _e('Error message for case 3', 'traveler');
38             }
39             ?>
40         </div>
41     }
42 }
```

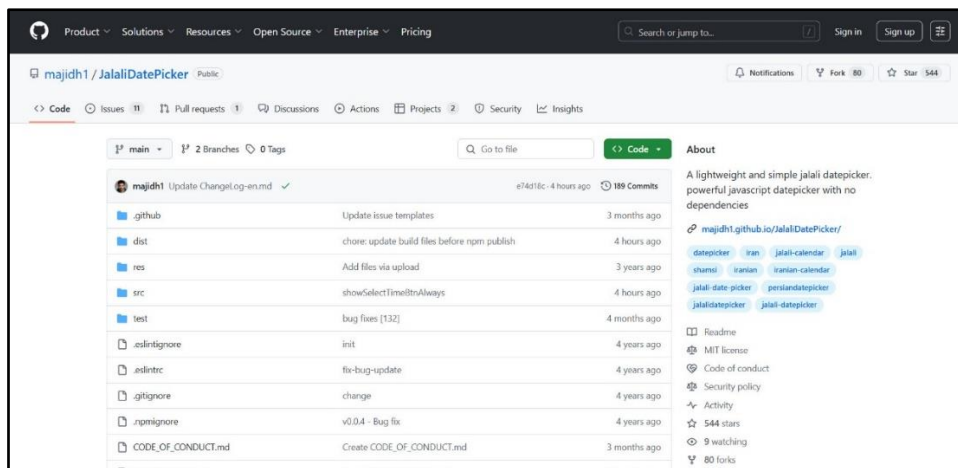
تصویر ۳-۸: مسدود کردن API‌های مربوط به لایسنس

۳-۵-۳ بررسی تقویم و جایگزینی فایل‌های daterangepicker و moment

یکی از اساسی‌ترین نیازهای بومی‌سازی برای کاربران فارسی‌زبان، تطبیق با تقویم شمسی (جلالی) و فرمت‌های تاریخ بومی است. بسیاری از قالب‌ها و افزونه‌ها از کتابخانه‌های جاوااسکریپت برای انتخاب و نمایش تاریخ استفاده می‌کنند. در قالب تراولر، کتابخانه‌های daterangepicker و moment.js برای مدیریت تاریخ و زمان مورد استفاده قرار گرفته بودند.

مراحل انجام شده برای شمسی‌سازی تقویم به شرح زیر بود:

- **بررسی تقویم برای اولین مرحله‌ی سفارشی‌سازی:** اولین قدم، شناسایی نقاطی در قالب بود که از تقویم میلادی استفاده می‌کردند. این شامل فیلدهای ورودی تاریخ، نمایش تاریخ در نتایج جستجو، و بخش‌های مربوط به رزرو بود.
- **پیدا کردن و جایگذاری daterangepicker با نسخه‌ی جلالی:**
 - **شناسایی فایل:** با جستجو در کدهای قالب، فایل‌ها و اسکریپت‌هایی که daterangepicker را فراخوانی می‌کردند، شناسایی شدند.
 - **دریافت نسخه‌ی جلالی:** یک نسخه‌ی فارسی‌سازی شده از daterangepicker که از تقویم شمسی پشتیبانی می‌کند (معمولاً با ترکیب با moment-jalaali.js) پیدا و دانلود شد.
 - **جایگذاری:** فایل‌های daterangepicker.js و daterangepicker.css اصلی قالب، با نسخه‌های جلالی آن‌ها جایگزین شدند. این کار معمولاً نیاز به بررسی دقیق نحوه‌ی فراخوانی اسکریپت‌ها در فایل functions.php یا فایل‌های دیگر قالب داشت تا اطمینان حاصل شود که نسخه‌های جدید به درستی بارگذاری می‌شوند.



تصویر ۳-۹: دریافت تقویم جلالی از گیت‌هاب

• جایگذاری moment جلالی درون پوسته:

- شناسایی فایل: کتابخانه‌ی moment.js که daterangepicker و سایر بخش‌های قالب به آن وابسته بودند، شناسایی شد.
- دریافت **moment-jalaali.js**: کتابخانه‌ی moment-jalaali.js که افزونه‌ای برای moment.js جهت پشتیبانی از تقویم شمسی است، دانلود شد.
- جایگذاری/افزودن: فایل moment.js اصلی با نسخه‌ی اصلی moment.js به همراه moment-jalaali.js (که باید پس از moment.js بارگذاری شود) جایگزین یا اضافه شدند. این کار به moment.js قابلیت‌های لازم برای تبدیل تاریخ‌های میلادی به شمسی و برعکس را می‌داد.
- پیکربندی منطق تاریخ: پس از جایگذاری فایل‌ها، نیاز به تغییراتی در کدهای جاوااسکریپت قالب بود تا daterangepicker از moment-jalaali برای نمایش تاریخ‌های شمسی استفاده کند. این شامل تنظیم locale یا calendar به jalaali در هنگام فراخوانی daterangepicker و تنظیمات moment.js برای کار با زبان فارسی بود.

۳-۵-۴) شمسی‌سازی کامل تقویم (js, css)

فرآیند شمسی‌سازی تنها به جایگذاری کتابخانه‌ها ختم نمی‌شد، بلکه شامل سفارشی‌سازی‌های عمیق‌تر در جاوااسکریپت و CSS برای اطمینان از نمایش صحیح و عملکرد کامل تقویم شمسی بود.

- ترجمه‌ی ماه‌های تقویم به فارسی: پس از جایگذاری کتابخانه‌های moment-jalaali و daterangepicker، نام ماه‌ها و روزهای هفته در تقویم به فارسی ترجمه شد. این کار معمولاً از طریق تنظیمات locale در moment.js یا از طریق رشته‌های ترجمه‌ی موجود در فایل‌های زبان daterangepicker انجام می‌گرفت.
- بررسی و سفارشی‌سازی کانفیگ daterangepicker: daterangepicker دارای گزینه‌های پیکربندی متعددی است که می‌توانند برای سفارشی‌سازی ظاهر و عملکرد آن به کار روند. در این مرحله، تنظیماتی مانند فرمت تاریخ، نمایش دکمه‌های تأیید و لغو، و حداقل/حداکثر تاریخ قابل انتخاب، برای تطابق با نیازهای تقویم شمسی و رابط کاربری فارسی بررسی و اصلاح شدند.
- بررسی منطق تنظیم اولیه‌ی تاریخ در پوسته: کدهای جاوااسکریپت مسئول منطق تنظیم تاریخ پیش‌فرض یا تاریخ رزرو در زمان بارگذاری صفحه بررسی شد و اطمینان حاصل گردید که تاریخ‌های اولیه نیز به درستی با تقویم شمسی نمایش داده می‌شوند و ورودی‌های کاربر نیز به فرمت شمسی دریافت می‌شوند.
- مشابه‌سازی تنظیمات نسخه‌ی سفارشی daterangepicker در نسخه‌ی جلالی: برخی قالب‌های حرفه‌ای، نسخه‌ی سفارشی‌سازی‌شده‌ای از daterangepicker را با تنظیمات خاص یا قابلیت‌های اضافه‌تر همراه داشتند. در این حالت، تنظیمات و منطق‌های کدنویسی شده در نسخه‌ی سفارشی قالب شناسایی و به نسخه‌ی جلالی daterangepicker انتقال داده شد تا قابلیت‌های اصلی قالب حفظ شوند. این شامل بررسی گزینه‌های فراخوانی daterangepicker و اطمینان از سازگاری آن‌ها با نسخه‌ی جلالی بود.
- مشابه‌سازی event listener های نسخه‌ی سفارشی daterangepicker در نسخه‌ی جلالی: قالب‌ها معمولاً event listener هایی را برای واکنش به تغییرات در انتخاب تاریخ (مانند انتخاب بازه‌ی تاریخ، اعمال فیلترها، و به‌روزرسانی قیمت‌ها) تعریف می‌کنند. این event listener ها با نسخه‌ی جلالی daterangepicker نیز سازگار شدند تا رویدادهای مربوط به تغییر تاریخ همچنان فعال باشند و داده‌ها به درستی به سایر بخش‌های قالب ارسال شوند.
- مشابه‌سازی پروتوتایپ‌های نسخه‌ی سفارشی daterangepicker (پارت ۱، ۲، ۳): daterangepicker یک کتابخانه‌ی مبتنی بر پروتوتایپ است و ممکن بود قالب، توابع یا ویژگی‌های جدیدی را به پروتوتایپ اصلی آن اضافه کرده باشد. این توابع سفارشی برای

عملکرد صحیح قالب حیاتی بودند. پروتوتایپ‌های شناسایی شده بررسی شدند تا اطمینان حاصل شود که در نسخه‌ی جلالی `daterangepicker` نیز به درستی کار می‌کنند. این مرحله شامل کپی برداری یا بازنویسی بخش‌هایی از کد برای سازگاری بود.

- **تعریف CSS مناسب و راست‌چین‌سازی `daterangepicker` جلالی:** علاوه بر تغییرات در `JavaScript`، ظاهر `daterangepicker` نیز برای زبان فارسی به درستی راست‌چین شد. این شامل تنظیم جهت متن و چیدمان عناصر داخل تقویم، معکوس کردن فلش‌های ناوبری (ماه قبل/بعد)، تنظیم حاشیه‌ها و فاصله‌ها برای نمایش صحیح اعداد و ماه‌ها و اعمال فونت‌های فارسی مناسب بود.
- رفع ایراد کنترلرهای رفتن به ماه قبل و بعد در نسخه‌ی جلالی: ایرادات مربوط به ناوبری بین ماه‌ها در نسخه‌های جلالی تقویم (مانند عدم تغییر ماه، یا تغییر به ماه اشتباه) شناسایی و منطق مربوط به این کنترلرها در جاوااسکریپت بررسی و اصلاح شد.
- رفع ایرادات نمایش تقویم تکی (`single datepicker`): اطمینان حاصل شد که نسخه‌ی جلالی `daterangepicker` در حالت `single datepicker` نیز به درستی عمل می‌کند و تمامی ایرادات مربوط به نمایش یا انتخاب تاریخ تکی برطرف می‌شوند.

```

4  .elementor-71 .elementor-element.elementor-element-2e4cc274 {
5      margin-top: 79px;
6      margin-bottom: 0px;
7  }
8  .elementor-element, .elementor-lightbox {
9      --swiper-theme-color: #000;
10     --swiper-navigation-size: 44px;
11     --swiper-pagination-bullet-size: 6px;
12     --swiper-pagination-bullet-horizontal-gap: 6px;
13 }
14 .elementor-element {
15     --widgets-spacing: 0px 0px;
16     --widgets-spacing-row: 0px;
17     --widgets-spacing-column: 0px;
18 }
19 .elementor-section {
20     position: relative;
21 }
22 .elementor-element {
23     --flex-direction: initial;
24     --flex-wrap: initial;
25     --justify-content: initial;
26     --align-items: initial;
27     --align-content: initial;
28     --gap: initial;
29     --flex-basis: initial;
30     --flex-grow: initial;
31     --flex-shrink: initial;
32     --order: initial;
33     --align-self: initial;
34     align-self: var(--align-self);
35     flex-basis: var(--flex-basis);
36     flex-grow: var(--flex-grow);
37     flex-shrink: var(--flex-shrink);
38     order: var(--order);
39 }
40 .elementor *, .elementor :after, .elementor :before {
41     box-sizing: border-box;

```

تصویر ۳-۱۰: فارسی سازی CSS

۳-۵-۵) افزودن jdatetime به قالب و سفارشی سازی و توسعه ی آن

در کنار شمس سازی در سمت کلاینت (جاوااسکریپت)، مدیریت تاریخ های شمس در سمت سرور (PHP) نیز برای ذخیره سازی، پردازش و نمایش تاریخ در پنل مدیریت یا در گزارش ها ضروری بود. کتابخانه ی jdatetime برای PHP، به عنوان ابزاری قدرتمند برای این منظور، به پروژه اضافه شد.

• افزودن jdatetime به قالب و سفارشی سازی و توسعه ی آن:

- نصب با Composer: jdatetime از طریق Composer به پروژه اضافه شد.

- **شناسایی نقاط نیاز به تبدیل تاریخ:** تمامی بخش‌های قالب که با تاریخ‌ها سروکار داشتند (مانند تاریخ رزرو، تاریخ انتشار، تاریخ پرداخت) شناسایی شدند.
- **تبدیل تاریخ‌های ورودی:** تاریخ‌های میلادی که از سمت کاربر دریافت می‌شدند، با استفاده از `strtotime` به تاریخ‌های شمسی تبدیل و ذخیره شدند.
- **نمایش تاریخ‌های خروجی:** تاریخ‌های ذخیره‌شده در پایگاه داده (که ممکن است به فرمت میلادی باشند) با استفاده از `strtotime` به فرمت شمسی تبدیل و برای نمایش به کاربر یا در پنل مدیریت آماده شدند.
- **سفارشی‌سازی فرمت نمایش `strtotime`:** قابلیت سفارشی‌سازی فرمت نمایش تاریخ (مانند "۱۴۰۴/۰۴/۱۵" یا "۱۵ تیر ۱۴۰۴") را فراهم می‌کند که بر اساس نیازهای قالب اعمال شد.

۳-۵-۶) افزودن قابلیت تبدیل تاریخ در منطق قالب به وسیله‌ی متود اختصاصی (PHP)

برای اطمینان از یکپارچگی و جلوگیری از تکرار کد، یک متود اختصاصی در قالب ایجاد شد تا تمامی تبدیل‌های تاریخ از میلادی به شمسی و برعکس را مدیریت کند.

- **تبدیل اولیه‌ی تاریخ میلادی به جلالی توسط `strtotime`:** در اولین گام، توابعی در فایل `functions.php` یا یک فایل کمکی در پوشه‌ی `includes` ایجاد شد تا تاریخ‌های میلادی ورودی را بلافاصله پس از دریافت (مثلاً از فرم‌های رزرو) با استفاده از `strtotime` به تاریخ شمسی تبدیل کرده و ذخیره کند. این کار اطمینان می‌داد که تمامی تاریخ‌ها در پایگاه داده به صورت استاندارد شمسی ثبت می‌شوند یا حداقل قابلیت تبدیل صحیح را دارند.
- **بازگردانی ویرایش‌های غیرضروری و رفع مشکلات `strtotime`:** در طول فرآیند بومی‌سازی، تغییرات یا کدهای اضافی که بعدها غیرضروری یا مشکل‌زا تشخیص داده شدند (مانند تداخل با سایر توابع تاریخ وردپرس)، شناسایی و بازگردانی شدند. همچنین، هرگونه باگ یا مشکل عملکردی که در یکپارچه‌سازی `strtotime` با منطق اصلی قالب پیش آمده بود، رفع شد.
- **پیدا کردن پردازنده‌ی تاریخ در منطق پوسته:** برای یکپارچه‌سازی کامل، تمامی توابع و کلاس‌هایی که در قالب وظیفه‌ی پردازش تاریخ (ذخیره، بازیابی، نمایش) را بر عهده داشتند، شناسایی شدند. این شامل توابعی بود که تاریخ‌ها را از فرم‌ها دریافت می‌کردند، آن‌ها را در دیتابیس ذخیره می‌کردند، یا از دیتابیس بازیابی و در صفحات نمایش می‌دادند.

- **تصحیح اولیه‌ی تبدیل تاریخ میلادی به جلالی در منطق:** پس از شناسایی پردازنده‌های تاریخ، تغییرات اولیه‌ای برای فراخوانی توابع `jdatetime` در این نقاط اعمال شد. هدف این بود که اطمینان حاصل شود تبدیل تاریخ‌ها از همان ابتدای فرآیند پردازش، به درستی انجام می‌شود.
- **تکمیل تبدیل تاریخ میلادی به جلالی با کدنویسی اختصاصی:** با توجه به پیچیدگی‌های قالب تراولر و نیاز به مدیریت تاریخ در سناریوهای مختلف (مثلاً محاسبه‌ی تعداد شب‌ها، بررسی در دسترس بودن اتاق‌ها بر اساس تاریخ شمسی)، نیاز به کدنویسی اختصاصی برای تکمیل فرآیند تبدیل تاریخ بود. این شامل ایجاد توابع کمکی برای تبدیل دوطرفه‌ی تاریخ‌ها و استفاده از آن‌ها در سراسر منطق قالب بود.
- **تعریف متود اختصاصی جهت تبدیل تاریخ در منطق پوسته:** برای حفظ کد تمیز و قابل نگهداری، یک متود (یا تابع) سراسری در فایل `functions.php` یا یک کلاس اختصاصی برای مدیریت تمامی تبدیل‌های تاریخ ایجاد شد. این متود، یک تاریخ میلادی را دریافت کرده و نسخه‌ی شمسی آن را برمی‌گرداند، یا بالعکس. این کار از تکرار کد جلوگیری کرده و مدیریت تاریخ را در یک مکان متمرکز می‌کند.

مثال کد ساده برای تعریف متود اختصاصی تبدیل تاریخ:

```

1  PHP
2  <?php
3  if (!function_exists('traveller_convert_to_jalali')) {
4      function traveller_convert_to_jalali($gregorian_date_string, $format = 'Y/m/d') {
5          try {
6              // Ensure jdatetime is loaded
7              if (!class_exists('jDateTime')) {
8                  // This is where you would typically autoload via Composer
9                  // For this example, we assume it's loaded.
10                 return $gregorian_date_string; // Fallback if jDateTime is not available
11             }
12
13             $jdate = new jDateTime(true, true, 'Asia/Tehran');
14             $parts = explode('-', $gregorian_date_string);
15             if (count($parts) === 3) {
16                 $jalali_array = $jdate->toJalali($parts[0], $parts[1], $parts[2]);
17                 return implode('/', $jalali_array); // Returns 'YYYY/MM/DD'
18             }
19             return $gregorian_date_string; // Return original if format is wrong
20         } catch (Exception $e) {
21             error_log('Error converting date: ' . $e->getMessage());
22             return $gregorian_date_string; // Return original on error
23         }
24     }
25 }

```

تصویر ۳-۱۱: متود اختصاصی تبدیل تاریخ

۳-۵-۶-۱) بررسی و توسعه‌ی درگاه پرداخت بومی

پس از حل مشکلات مربوط به تاریخ، یکپارچگی با سیستم‌های پرداخت بومی، به خصوص برای قالب‌های فروشگاه‌های یا رزرو، به عنوان یک جنبه‌ی حیاتی بومی‌سازی مورد بررسی قرار گرفت. قالب تراولر به صورت پیش‌فرض از درگاه‌های پرداخت بین‌المللی مانند PayPal پشتیبانی می‌کرد. نیاز بود که این سیستم با درگاه‌های پرداخت ایرانی جایگزین شود و واحد پول ریال یا تومان برای سرویس‌های قالب تعریف گردد.

- **بررسی صفحه پرداخت و افزونه‌ی PayPal:** صفحه‌ی پرداخت در قالب و نحوه‌ی عملکرد افزونه‌ی PayPal به دقت بررسی شد. این بررسی شامل تحلیل ساختار فرم‌های پرداخت، نحوه‌ی ارسال اطلاعات به سمت سرور، و روش ارتباط با API درگاه پرداخت بود. این شناخت، مبنایی برای توسعه‌ی درگاه پرداخت ایرانی فراهم آورد.
- **کپی‌برداری از افزونه‌ی پرداخت PayPal جهت توسعه:** برای شروع توسعه‌ی درگاه پرداخت ایرانی، افزونه‌ی PayPal که همراه قالب بود، به عنوان یک نمونه‌ی اولیه کپی‌برداری شد. این اقدام امکان استفاده از ساختار و توابع موجود در افزونه‌ی اصلی را فراهم آورد و از شروع کار از صفر جلوگیری کرد. با کپی‌برداری، یک افزونه‌ی جدید با نام مربوط به درگاه پرداخت ایرانی ایجاد شد که امکان اعمال تغییرات لازم در آن وجود داشت.
- **شروع توسعه‌ی افزونه‌ی پرداخت برای سفارش‌های درون پوسته:** توسعه‌ی افزونه‌ی پرداخت جدید شامل تعریف کلاس‌ها و توابع لازم برای ارتباط با API درگاه پرداخت ایرانی، پیاده‌سازی متدهای ارسال درخواست پرداخت و دریافت پاسخ از درگاه، و هندل کردن Callback ها و به‌روزرسانی وضعیت سفارش در پایگاه داده وردپرس بود.
- **رفع خطا و دیباگ افزونه‌ی پرداخت IDPay:** در طول توسعه‌ی افزونه‌ی پرداخت IDPay، خطاهای متعددی (مانند خطاهای ارتباط با API، مشکلات در ارسال پارامترها، یا مشکلات در دریافت پاسخ) بروز کردند. برای رفع این خطاها، از ابزارهای دیباگینگ (مانند dd() و لاگ‌های PHP) استفاده شد. این فرآیند شامل بررسی مستندات IDPay API، تست سناریوهای مختلف پرداخت و اطمینان از صحت تراکنش‌ها بود.
- **تکمیل افزونه‌ی درگاه پرداخت اختصاصی قالب به وسیله‌ی IDPay:** در نهایت، تمامی قابلیت‌های لازم برای یک درگاه پرداخت کامل، از جمله نمایش پیغام‌های موفقیت/شکست،

مدیریت خطاهای کاربرپسند، و یکپارچگی کامل با سیستم سفارش‌دهی قالب، تکمیل شد. هدف از این بخش، ارائه‌ی یک تجربه‌ی پرداخت روان و بدون مشکل برای کاربران ایرانی بود.

۳-۵-۷) ترجمه محتوای استاتیک و پست‌تایپ‌های قالب

پس از انجام بومی‌سازی فنی (تقویم و پرداخت)، نوبت به ترجمه‌ی متون قالب رسید. این شامل ترجمه‌ی تمامی رشته‌های متنی بود که در بخش‌های مختلف قالب، از جمله محتوای استاتیک، پیام‌ها، و لیبل‌های پست‌تایپ‌ها، نمایش داده می‌شدند.

• ترجمه‌ی محتوای استاتیک و پست‌تایپ‌های قالب:

- استخراج رشته‌های قابل ترجمه: با استفاده از ابزارهایی مانند Poedit یا Loco Translate، تمامی رشته‌های متنی که با توابع ترجمه‌ی وردپرس (مانند __() و __e()) در قالب کدنویسی شده بودند، استخراج شدند.
- ترجمه: این رشته‌ها به دقت به فارسی ترجمه شدند. در فرآیند ترجمه، علاوه بر صحت معنایی، به جنبه‌های فرهنگی و اصطلاحات رایج در زبان فارسی نیز توجه شد تا متن نهایی کاملاً طبیعی و بومی به نظر برسد.
- ترجمه‌ی پست‌تایپ‌ها و فیلدهای سفارشی: قالب تراولر از پست‌تایپ‌های سفارشی (مانند "Car", "Hotel", "Tour") و فیلدهای سفارشی متعددی برای اطلاعات رزرو استفاده می‌کرد. لیبل‌ها، نام‌ها و توضیحات این پست‌تایپ‌ها و فیلدها نیز به فارسی ترجمه شدند تا پنل مدیریت برای کاربران فارسی‌زبان کاملاً قابل فهم باشد.

۳-۵-۸) بررسی فایل‌های ترجمه و تکنیک‌های مرتبط

مدیریت صحیح فایل‌های ترجمه (با پسوند po و mo) و استفاده از تکنیک‌های بهینه، برای حفظ کیفیت و سهولت به‌روزرسانی ترجمه‌ها حیاتی بود.

• کارکردن و استفاده از فایل‌های ترجمه:

- استاندارد نام‌گذاری: فایل‌های po و mo با نام‌گذاری استاندارد fa_IR.po و fa_IR.mo (برای زبان فارسی-ایران) در پوشه‌ی languages قالب یا افزونه ذخیره شدند.

- به‌روزرسانی ترجمه‌ها: در صورت انتشار نسخه‌ی جدید قالب، فایل POT. ممکن بود به‌روزرسانی شود. در این صورت، با استفاده از Poedit، فایل po. موجود با فایل POT. جدید مقایسه شده و رشته‌های جدید اضافه و ترجمه می‌شدند.
- اولویت‌بندی بارگذاری ترجمه: وردپرس ابتدا فایل‌های ترجمه را از پوشه‌ی languages درون قالب/افزونه می‌خواند. اگر فایل‌های ترجمه در این مسیر نباشند، به پوشه‌ی wp-content/languages رجوع می‌کرد.
- نکات کیفی: استفاده از واژگان یکنواخت، رعایت نیم‌فاصله، و استفاده از علائم نگارشی صحیح در ترجمه، برای حفظ کیفیت متن بومی‌شده ضروری بود.

۳-۵-۹ بررسی فایل CSS راست‌چین و افزودن فونت

استایل‌دهی و راست‌چین‌سازی CSS در کنار مدیریت فونت‌ها، از دیگر جنبه‌های مهم بومی‌سازی بود که بر ظاهر و تجربه‌ی کاربری تأثیر مستقیم داشت.

• بررسی فایل CSS راست‌چین و افزودن فونت:

- شناسایی فایل‌های CSS: تمامی فایل‌های CSS اصلی قالب و همچنین فایل rtl.css (در صورت وجود) بررسی شدند.
- اعمال تغییرات RTL: تغییرات لازم برای راست‌چین‌سازی (مانند direction: rtl و text-align: right, align: right, float ها و margin/padding ها) در فایل rtl.css (اگر قالب از آن پشتیبانی می‌کرد) یا در فایل style.css (با استفاده از سلکتور [html[dir="rtl"]]) اعمال شد.
- افزودن فونت‌های فارسی: برای نمایش زیبا و خوانایی متن فارسی، فونت‌های فارسی مناسب (مانند ایران‌سنس، وزیر، یا گندم) به قالب اضافه و در CSS فراخوانی شدند. این کار شامل وارد کردن فایل‌های فونت (.woff ، .woff2 ، .ttf و غیره) در پوشه‌ی assets/fonts یا css/fonts قالب، تعریف @font-face در CSS برای هر فونت و اعمال font-family به عناصر متنی اصلی بود.

۳-۵-۱۰) سناریو تعریف فونت در ریداکس

در برخی قالب‌های پیچیده (مانند قالب تراولر)، تنظیمات گسترده‌ای از طریق فریم‌ورک‌های آپشن پنل (مانند Redux Framework) مدیریت می‌شوند. این فریم‌ورک‌ها امکان می‌دهند تا کاربر از طریق پنل مدیریت، گزینه‌های مختلف قالب از جمله فونت را تغییر دهد.

• سناریو تعریف فونت در ریداکس:

- **شناسایی گزینه‌های فونت:** در پنل تنظیمات قالب تراولر که مبتنی بر Redux بود، گزینه‌های مربوط به تنظیم فونت شناسایی شدند.
- **افزودن فونت‌های فارسی به گزینه‌ها:** برای اینکه کاربر بتواند به راحتی از میان فونت‌های فارسی انتخاب کند، نام و مسیر فونت‌های فارسی به لیست فونت‌های قابل انتخاب در Redux اضافه شد. این کار نیاز به ویرایش فایل‌های PHP مربوط به تعریف آپشن پنل Redux در قالب داشت.
- **اعمال فونت از طریق PHP و CSS:** پس از انتخاب فونت توسط کاربر در پنل مدیریت، Redux این انتخاب را در پایگاه داده ذخیره می‌کرد. سپس، در فایل functions.php یا یک فایل کمکی، منطقی نوشته شد تا بر اساس انتخاب کاربر، CSS لازم برای فراخوانی آن فونت را تولید کرده و به سربرگ وبسایت اضافه کند. این کار به قالب امکان می‌داد تا فونت‌های فارسی را به صورت دینامیک بارگذاری کند.

۳-۶) حل مشکلات مربوط به راست‌چین‌سازی (CSS و فایل‌های قالب)

این بخش به خلاصه‌سازی چالش‌های کلی و راهکارهای جامع در راست‌چین‌سازی قالب، فراتر از موارد ذکر شده برای تقویم و فونت، می‌پردازد. بسیاری از این چالش‌ها در طول فرآیند مستندسازی (بند ۳-۵) به صورت عملی رفع شدند.

• مشکلات رایج CSS در RTL

- **تداخل با فریم‌ورک‌های UI:** برخی قالب‌ها از فریم‌ورک‌های CSS مانند Bootstrap یا Foundation استفاده می‌کردند که ممکن بود نیاز به نسخه‌ی RTL خاص خود داشته باشند یا نیاز به بازنویسی قوانین CSS برای سازگاری با RTL داشته باشند.

- **آیکون‌ها و فلش‌ها:** یکنواختی و فلش‌هایی که جهت‌دار هستند (مانند فلش‌های ناوبری یا آیکون‌های مرتب‌سازی) باید برای RTL معکوس می‌شدند. این کار معمولاً با تغییر transform: scaleX(-1); یا جایگزینی آیکون انجام می‌شد.
- **جدول‌ها و فرم‌ها:** چینش ستون‌ها در جدول‌ها و ترتیب فیلدها در فرم‌ها ممکن بود نیاز به تنظیم مجدد داشته باشد تا از راست به چپ صحیح نمایش داده شوند.
- **Pop-up ها و Modal ها:** موقعیت‌دهی این عناصر در صفحه‌ی راست‌چین باید تنظیم می‌شد تا در جای مناسب و با جهت صحیح نمایش داده شوند.

• مشکلات در فایل‌های قالب (HTML/PHP)

- **ویژگی dir:** اطمینان حاصل شد که تگ <html> دارای ویژگی dir="rtl" در صفحات فارسی است.
- **چینش عناصر:** برخی عناصر HTML ممکن بود نیاز به تغییر ترتیب داشته باشند تا در RTL به درستی چیده شوند (مثلاً سایدبارها).
- **تصاویر و گرافیک‌ها:** تصاویر یا گرافیک‌هایی که دارای متن یا جهت خاصی هستند، ممکن بود نیاز به نسخه‌ی RTL داشته باشند.

برای حل این مشکلات، رویکرد اصلی، بررسی دقیق هر بخش از قالب در حالت RTL، شناسایی خطاهای بصری، و اعمال تغییرات لازم در فایل‌های CSS مربوطه بود. استفاده از ابزارهای مرورگر (Developer Tools) برای بازرسی عناصر و آزمایش تغییرات CSS به صورت زنده، در این فرآیند بسیار مؤثر بود.

۳-۷) تست نهایی و آماده‌سازی برای انتشار

پس از اتمام تمامی مراحل بومی‌سازی فنی و ترجمه، مرحله‌ی نهایی شامل تست جامع قالب و آماده‌سازی آن برای انتشار در مارکت‌پلیس‌های ایرانی بود.

۳-۷-۱) ایجاد پیش‌نمایش پوسته درون هاست

قبل از ارسال قالب به مارکت‌پلیس، لازم بود یک پیش‌نمایش (Demo) زنده از آن در یک محیط هاست واقعی ایجاد شود. این پیش‌نمایش به خریداران بالقوه امکان می‌دهد تا قالب بومی‌سازی شده را در عمل مشاهده و عملکرد آن را بررسی کنند.

مراحل ایجاد پیش‌نمایش:

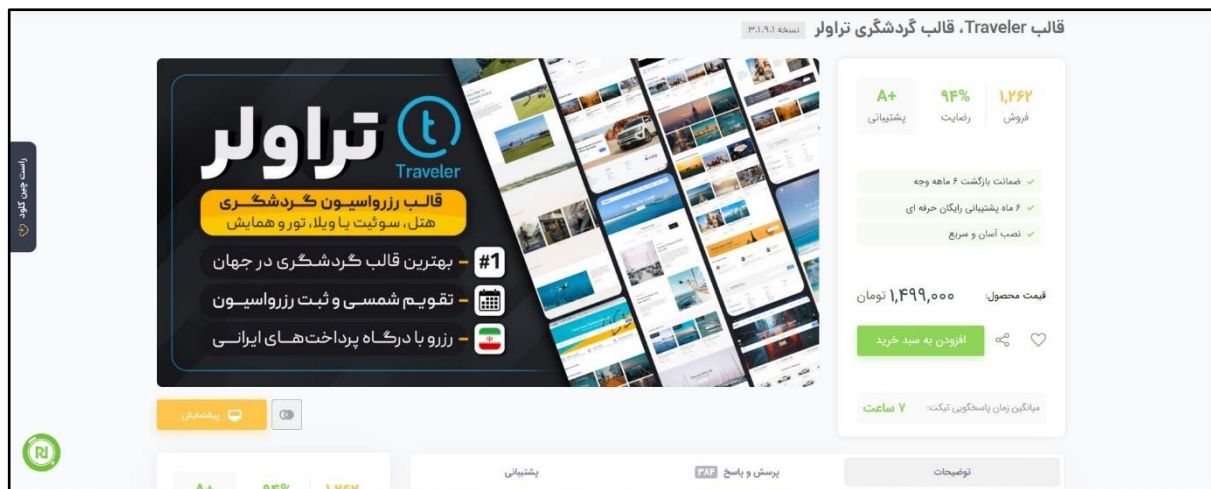
- **انتخاب هاست مناسب:** یک هاست با مشخصات مناسب و سازگار با وردپرس انتخاب شد.
- **انتقال قالب و اطلاعات:** وبسایت وردپرسی حاوی قالب بومی‌سازی شده (که در محیط Laragon توسعه یافته بود) با استفاده از افزونه‌هایی مانند Duplicator یا به صورت دستی (کپی کردن فایل‌ها و انتقال پایگاه داده)، به هاست منتقل شد.
- **پیکربندی دامنه‌ی پیش‌نمایش:** یک ساب‌دامین (مثلاً demo.yourdomain.com) برای پیش‌نمایش ایجاد شد و وردپرس در آن پیکربندی شد.
- **تست جامع:** تمامی بخش‌های قالب (صفحات، پست‌ها، فرم‌ها، سیستم رزرو، درگاه پرداخت) در محیط هاست واقعی مورد تست قرار گرفتند تا از عملکرد صحیح در محیط واقعی و سازگاری کامل با سرور اطمینان حاصل شود.
- **بارگذاری محتوای نمونه:** برای جذابیت بیشتر پیش‌نمایش، محتوای نمونه‌ی مرتبط با قالب (مانند تصاویر، متون، لیست هتل‌ها/تورها) به آن اضافه شد تا کاربران بتوانند قابلیت‌های قالب را به طور کامل مشاهده کنند.

۳-۸) ثبت در مارکت ایرانی (مطالعه موردی: راست‌چین)

آخرین گام، آماده‌سازی محصول و ارسال آن به مارکت‌پلیس انتخابی برای بررسی و انتشار بود. در این مطالعه‌ی موردی، با توجه به الزامات رایج، بر فرآیند ارسال به راست‌چین تمرکز شد.

۳-۸-۱) ایجاد لندینگ و محتوای متناسب برای انتشار در راست‌چین

برای موفقیت یک محصول در مارکت‌پلیس، علاوه بر کیفیت فنی، ارائه‌ی جذاب و مستندات کامل نیز حیاتی است. مارکت‌پلیس‌ها معمولاً راهنماهایی برای این بخش ارائه می‌دهند.



تصویر ۳-۱۲: پیش نمایش قالب روی سایت راست چین

• محتوای متنی در راست چین:

- **عنوان و توضیحات محصول:** یک عنوان جذاب و توضیحات جامع و دقیق در مورد قابلیت‌ها، ویژگی‌ها و مزایای قالب بومی‌سازی شده تهیه شد. این توضیحات باید شفاف و بدون ابهام باشند.
- **لیست قابلیت‌ها:** تمامی قابلیت‌های اصلی و بومی‌سازی شده قالب (مانند پشتیبانی از تقویم شمسی، درگاه پرداخت ایرانی، راست‌چین‌سازی کامل) به صورت لیست و با جزئیات ذکر شدند.
- **نحوه‌ی نصب و راه‌اندازی:** راهنمای گام به گام نصب و راه‌اندازی قالب (برای کاربران) تهیه شد.
- **پشتیبانی و به‌روزرسانی:** اطلاعات مربوط به نحوه‌ی پشتیبانی (زمان، کانال‌ها) و برنامه‌ی به‌روزرسانی‌های آینده‌ی قالب درج شد.

• محتوای تصویری در راست چین:

- **اسکرین‌شات‌های با کیفیت:** چندین اسکرین‌شات با کیفیت بالا از بخش‌های مختلف قالب در حالت بومی‌سازی شده (صفحات اصلی، پنل مدیریت، فیلم‌های تاریخ شمسی، صفحات پرداخت) تهیه شد.
- **تصاویر پیش‌نمایش:** تصاویری جذاب و با وضوح بالا برای صفحه‌ی محصول در مارکت پلیس آماده شد.
- **ویدئوهای معرفی (اختیاری):** در صورت امکان، یک ویدئوی کوتاه معرفی از قابلیت‌های اصلی قالب و نحوه‌ی عملکرد آن نیز تهیه شد.

- رعایت اندازه‌ها و کیفیت‌های استاندارد: تمامی تصاویر و ویدئوها باید از نظر ابعاد و کیفیت با استانداردهای مارکت پلیس (مثلاً راست‌چین) مطابقت داشته باشند.



تصویر ۳-۱۳: ایجاد اسکرین شات‌های با کیفیت در صفحه‌ی لندینگ

۳-۸-۲) ارسال به راست‌چین (تمامی مراحل + لایسنس‌گذاری روی قالب)

فرآیند ارسال محصول به مارکت پلیس راست‌چین شامل چندین مرحله است که نیاز به دقت و رعایت تمامی الزامات دارد.

• مراحل ارسال:

- **ثبت‌نام به عنوان فروشنده:** در ابتدا، در مارکت پلیس راست‌چین به عنوان فروشنده ثبت‌نام شد و اطلاعات لازم (مانند اطلاعات بانکی) تکمیل گردید.
- **ایجاد محصول جدید و بارگذاری فایل‌ها:** در پنل فروشندگی، گزینه‌ی "افزودن محصول جدید" انتخاب و فایل فشرده‌ی قالب (شامل قالب اصلی و Child Theme)، مستندات و تمامی فایل‌های مورد نیاز دیگر بارگذاری شد.

- **وارد کردن اطلاعات محصول:** تمامی جزئیات محصول شامل عنوان، توضیحات، دسته‌بندی، تگ‌ها، قیمت و لینک پیش‌نمایش وارد شدند.
- **بارگذاری تصاویر و محتوای معرفی:** اسکرین‌شات‌ها و تصاویر معرفی که پیشتر آماده شده بودند، در این بخش بارگذاری شدند.
- **لایسنس‌گذاری روی قالب:** سیستم لایسنس‌گذاری مخصوص مارکت‌پلیس بر روی قالب اعمال شد. این مرحله شامل تنظیمات لایسنس (مثلاً برای هر نصب یا به صورت تعداد محدود) بود تا پس از خرید، لایسنس به مشتری ارائه شود و قابلیت به‌روزرسانی و پشتیبانی فراهم آید. این لایسنس می‌توانست توسط خود مارکت‌پلیس مدیریت شود یا نیاز به یک افزونه‌ی لایسنس اختصاصی در قالب داشته باشد.
- **ارسال برای بررسی و رفع ایرادات:** پس از تکمیل تمامی اطلاعات، محصول برای بررسی توسط تیم راست‌چین ارسال شد. این مرحله شامل بررسی فنی، امنیتی، و کیفی محصول از نظر رعایت استانداردهای بومی‌سازی بود. در صورت وجود ایراد یا نیاز به اصلاح، تیم راست‌چین بازخورد خود را ارائه می‌داد و فروشنده اصلاحات لازم را انجام داده و مجدداً محصول را برای بررسی ارسال می‌کرد. پس از تأیید نهایی، محصول در مارکت‌پلیس منتشر می‌شد.

۳-۸-۳) کسب درآمد از فروش قالب در مارکت‌پلیس‌های ایرانی

عرضه‌ی محصولات بومی‌سازی‌شده در مارکت‌پلیس‌هایی نظیر راست‌چین، فرصت‌های قابل توجهی برای توسعه‌دهندگان به منظور کسب درآمد فراهم می‌آورد. راست‌چین به عنوان پلتفرمی برای اتصال توسعه‌دهندگان به کاربران فارسی‌زبان عمل می‌کند. [۲]

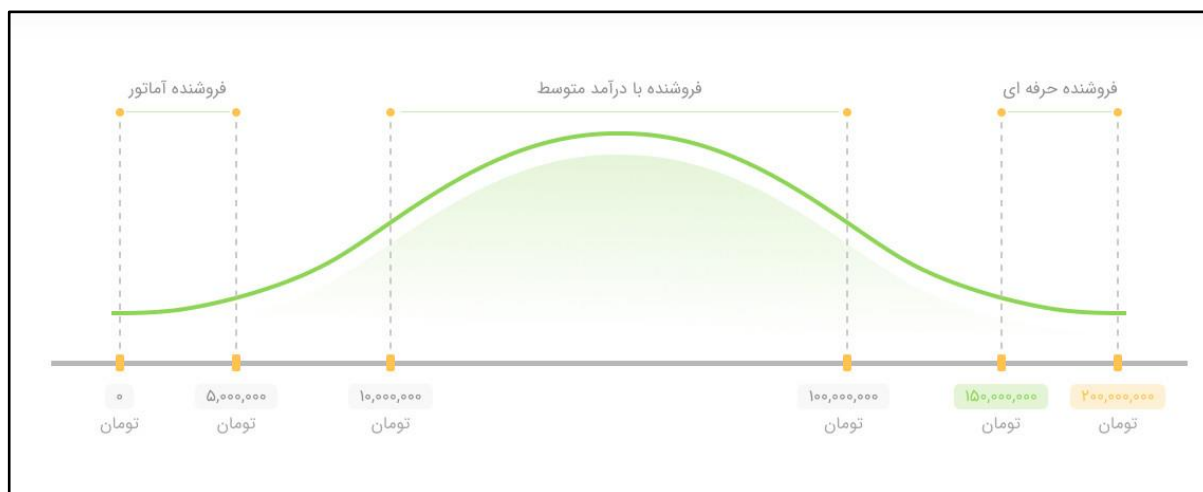
دو رویکرد اصلی برای کسب درآمد در این پلتفرم امکان‌پذیر است:

- **طراحی و عرضه‌ی محصولات اورجینال ایرانی:** توسعه‌دهندگان می‌توانند قالب‌ها یا افزونه‌هایی را از ابتدا با تمرکز بر نیازهای بازار ایران طراحی و در این مارکت‌پلیس عرضه کنند.
- **ترجمه و انتشار محصولات اورجینال (بین‌المللی):** این رویکرد، همان فرآیند بومی‌سازی است که محور اصلی این پایان‌نامه بود. با ترجمه و تطبیق قالب‌ها و افزونه‌های بین‌المللی با نیازهای کاربران ایرانی، امکان عرضه‌ی آن‌ها در راست‌چین فراهم می‌شود.

برای افزایش موفقیت در فروش و به تبع آن کسب درآمد، راست‌چین توصیه‌هایی را برای فروشندگان خود ارائه می‌دهد:

- **تحلیل نیازهای بازار:** پیش از توسعه یا بومی‌سازی، نیازهای واقعی کاربران ایرانی شناسایی شود تا محصولی با تقاضای بالا عرضه گردد.
- **تعریف استراتژی محصول:** یک استراتژی مشخص برای محصول (شامل ویژگی‌ها، جامعه هدف، قیمت‌گذاری و...) تعریف شود.
- **مشاوره و آموزش:** امکان بهره‌برداری از مشاوره‌ی رایگان تیم پشتیبانی راست‌چین و دنبال کردن دوره‌های آموزشی آکادمی آن‌ها برای ارتقای دانش و مهارت‌های لازم برای موفقیت در فروش فراهم است.

درآمد حاصل از فروش معمولاً به صورت درصدی بین فروشنده و مارکت‌پلیس تقسیم می‌شود که می‌تواند منبع درآمد پایدار برای توسعه‌دهندگان فعال باشد. این فرصت، اهمیت بومی‌سازی حرفه‌ای و با کیفیت را بیش از پیش نمایان می‌سازد.



تصویر ۳-۱۴: درآمد فروشندگان راست چین

حسین اسماعیلیان

قالب فروشگاه‌های پارس کالا

تعداد کل فروش

۳,۰۲۵

درآمد کل

۱.۹ میلیارد تومان

درآمد ماهیانه

۱۳۰ میلیون تومان

مشاهده محصولات



PARS KALA

پارس کالا

حرفه‌ای‌ترین قالب
فروشگاهی و ووکامرس

تصویر ۳-۱۵: فروشندگان راست چین

محمد یزدانی

قالب ووودمارت

تعداد کل فروش

۲۴,۸۰۰

درآمد کل

۱.۵ میلیارد تومان

درآمد ماهیانه

۲۰۰ میلیون تومان

مشاهده محصولات



قالب و اپلیکیشن فروشگاهی

Woodmart

ووودمارت

Astra

آسترا

تصویر ۳-۱۶: فروشندگان راست چین

قالب فروشگاه ایرانی نگارشاپ

تعداد فروش

۵۱۰۷

میانگین درآمد ماهیانه

۲۰,۵۷۲,۵۴۳ تومان

کل درآمد فروش

۱,۴۸۱,۲۲۳,۱۰۸ تومان

قیمت محصول:

۱,۵۱۵,۰۰۰ تومان

مشاهده محصول



نگارشی دهم (۱۴۰۳)

طراحی حرفه‌ای برای انواع سلیقه‌ها

- تجهیزات قالب حرفه‌ای و آسان
- دموهای حرفه‌ای و چند صفحه‌ای
- ایزاک های حرفه‌ای المنتوری
- امکان چند فروشندگی دکان
- طراحی حرفه‌ای واکتشر + ایکسکش PWA

تصویر ۳-۱۷: پرفروش‌ترین قالب‌های راست چین



تصویر ۳- ۱۸: پرفروش ترین قالب های راست چین

فصل ۴: نتیجه‌گیری و پیشنهادات کارهای آتی

در این فصل، به جمع‌بندی یافته‌ها و اقدامات صورت‌گرفته در راستای بومی‌سازی قالب‌ها و افزونه‌های وردپرس برای مارکت‌پلیس‌های ایرانی می‌پردازیم. پس از آن، چالش‌های عمده‌ای که در طول این مسیر با آن‌ها مواجه شدیم، تشریح خواهند شد و در نهایت، پیشنهاداتی برای تحقیقات و اقدامات آتی در این حوزه ارائه می‌گردد. هدف از این فصل، ارائه یک دیدگاه جامع از دستاوردهای تحقیق و همچنین ترسیم افق‌های جدید برای توسعه‌ی اکوسیستم وردپرس فارسی است.

۴-۱) جمع‌بندی اقدامات انجام‌شده

این پایان‌نامه با هدف بررسی و مستندسازی فرآیند جامع بومی‌سازی قالب‌ها و افزونه‌های وردپرس برای عرضه در مارکت‌پلیس‌های ایرانی، گام‌های متعددی را از مباحث نظری تا پیاده‌سازی عملی طی کرده است. در فصل اول، به کلیات تحقیق، شامل مقدمه، بیان مسأله، اهمیت و ضرورت، اهداف، سؤالات، روش تحقیق و تعریف واژگان کلیدی پرداختیم تا چارچوبی نظری برای مطالعه فراهم آید. در فصل دوم، مروری بر ادبیات تحقیق و پیشینه‌ی نظری، شامل معرفی وردپرس، ساختار قالب‌ها و افزونه‌ها، مفهوم بومی‌سازی و معرفی مارکت‌پلیس‌های ایرانی نظیر ژاکت و راست‌چین، ارائه شد.

بخش عملی و اصلی این تحقیق در فصل سوم به تفصیل بیان گردید. در این فصل، فرآیند بومی‌سازی یک قالب مطرح بین‌المللی، یعنی قالب تراولر (Traveler)، به صورت عملی انجام شد. این فرآیند با انتخاب و خرید قالب از ThemeForest آغاز گردید و نکات مهمی در این خصوص بررسی شد. سپس، یک محیط توسعه‌ی محلی با استفاده از Laragon راه‌اندازی و وردپرس در آن نصب شد. این مرحله، بستر لازم برای کار بر روی قالب را فراهم آورد.

پس از نصب قالب و افزونه‌های مورد نیاز آن، اقدامات بومی‌سازی به شرح زیر پیگیری شد:

- **بررسی و جایگزینی فایل‌های تاریخ‌نگار:** این گام شامل بررسی عملکرد تقویم در قالب و جایگزینی کتابخانه‌های daterangepicker و moment با نسخه‌های جلالی آن‌ها بود.
- **شمسی‌سازی کامل تقویم و ترجمه ماه‌ها:** این مرحله فراتر از جایگزینی ساده بود و شامل ترجمه ماه‌های تقویم به فارسی، سفارشی‌سازی کانفیگ daterangepicker، بررسی منطق تنظیم اولیه تاریخ در پوسته و همچنین مشابه‌سازی تنظیمات، event listener ها و پروتوتایپ‌های نسخه‌ی سفارشی daterangepicker در نسخه‌ی جلالی می‌شد.

- تعریف CSS مناسب و راست‌چین‌سازی **daterangepicker** جلالی: این بخش شامل اصلاحات ظاهری برای نمایش صحیح تقویم جلالی در جهت راست‌به‌چپ بود.
- رفع ایرادات نمایش تقویم تکی (**single datepicker**) و کنترل‌رهای ماه قبل/بعد: رفع مشکلات جزئی اما مهم در نمایش تقویم‌ها که تجربه کاربری را بهبود می‌بخشد.
- افزودن **composer** و **dd** برای دیباگ: استفاده از ابزارهای توسعه مانند **composer** و تابع **dd** (**dump and die**) برای تسهیل فرآیند دیباگ و عیب‌یابی در طول پروژه.
- افزودن کتابخانه **jdatetime**: این کتابخانه PHP به طور خاص برای کار با تاریخ شمسی در PHP به قالب اضافه و سفارشی‌سازی شد.
- تبدیل تاریخ در منطق قالب با متود اختصاصی **PHP**: فرآیند تبدیل تاریخ میلادی به جلالی در لایه منطق قالب با استفاده از **jdatetime** و کدنویسی اختصاصی پیاده‌سازی شد. این شامل تبدیل اولیه، بازگردانی ویرایش‌های غیرضروری و رفع مشکلات **jdatetime**، پیدا کردن پردازنده تاریخ در منطق پوسته، تصحیح اولیه تبدیل، تکمیل تبدیل با کدنویسی اختصاصی و تعریف متود اختصاصی بود.
- شروع بررسی درگاه پرداخت درونی قالب و توسعه افزونه پرداخت اختصاصی: این بخش شامل بررسی ساختار پرداخت قالب (مانند **PayPal**) و کپی‌برداری و توسعه یک افزونه پرداخت اختصاصی برای درگاه‌های ایرانی مانند **IDPay** بود.
- ترجمه محتوای استاتیک و پست‌تایپ‌ها: تمامی رشته‌های متنی، شامل محتوای استاتیک و محتوای مربوط به پست‌تایپ‌های سفارشی قالب، به زبان فارسی ترجمه شدند.
- استفاده از فایل‌های ترجمه: بررسی و به‌کارگیری صحیح فایل‌های **PO** و **MO** برای مدیریت ترجمه‌ها.
- راست‌چین‌سازی CSS و افزودن فونت: انجام تغییرات لازم در فایل‌های CSS برای راست‌چین‌سازی کامل قالب و افزودن فونت‌های فارسی مناسب، از جمله سناریو تعریف فونت در ریداکس.
- آماده‌سازی برای انتشار در راست‌چین: این مرحله شامل ایجاد محتوای متنی و تصویری متناسب برای صفحه محصول در مارکت‌پلیس راست‌چین بود.
- ایجاد و اجرای پیش‌نمایش پوسته: راه‌اندازی یک نسخه پیش‌نمایش (**Demo**) از قالب بومی‌سازی‌شده بر روی هاست.

- **ارسال به راست‌چین:** نهایتاً، تمامی مراحل لازم برای ارسال قالب به مارکت‌پلیس راست‌چین، از جمله لایسنس‌گذاری، تکمیل و اجرا شد.

این اقدامات نشان داد که فرآیند بومی‌سازی یک محصول وردپرسی بین‌المللی، فراتر از ترجمه صرف بوده و نیازمند دانش فنی عمیق در زمینه‌های مختلف مانند JavaScript، PHP، CSS و همچنین شناخت دقیق ساختار وردپرس و نیازهای بازار بومی است.

۴-۲) چالش‌های encountered در پروژه

در طول فرآیند بومی‌سازی قالب و آماده‌سازی آن برای عرضه در مارکت‌پلیس‌های ایرانی، با چالش‌ها و مشکلات متعددی مواجه شدیم که شناخت آن‌ها برای توسعه‌دهندگان آتی ضروری است. این چالش‌ها را می‌توان به چند دسته اصلی تقسیم کرد:

- **چالش‌های فنی مربوط به بومی‌سازی تاریخ و زمان (تقویم شمسی):**

- عدم پشتیبانی ذاتی: اکثر قالب‌ها و افزونه‌های بین‌المللی به طور پیش‌فرض هیچ پشتیبانی از تقویم‌های غیرمیلادی ندارند. این امر نیاز به جایگزینی کامل کتابخانه‌های تاریخ‌نگار (مانند moment.js و daterangepicker.js) با نسخه‌های جلالی آن‌ها را ایجاد می‌کند.
- پیچیدگی یکپارچه‌سازی: یکپارچه‌سازی نسخه‌های جلالی این کتابخانه‌ها با منطق قالب، به دلیل تفاوت در ساختار توابع و نحوه فراخوانی، نیازمند زمان و تلاش زیادی بود. به خصوص، مشابه‌سازی event listener ها و پروتوتایپ‌های اختصاصی که قالب از نسخه اصلی daterangepicker استفاده می‌کرد، یکی از چالش‌های مهم بود.
- تغییرات در منطق PHP: علاوه بر JavaScript، منطق سمت سرور (PHP) نیز نیاز به تغییرات گسترده‌ای برای تبدیل تاریخ میلادی به جلالی داشت که با استفاده از کتابخانه‌ای مانند jdatetime و توسعه توابع اختصاصی صورت گرفت. این تغییرات باید به گونه‌ای انجام می‌شد که عملکرد اصلی قالب دستخوش اختلال نشود.
- مشکلات ظاهری: راست‌چین‌سازی تقویم‌های جاوااسکریپتی نیاز به تنظیمات دقیق CSS داشت تا چینش اعداد، روزها و ماه‌ها به درستی نمایش داده شود.

- چالش‌های مربوط به راست‌چین‌سازی (RTL):

- وابستگی به CSS: RTL به شدت به کدهای CSS وابسته است و در بسیاری از موارد، قالب‌های بین‌المللی CSS کافی یا بهینه‌ای برای RTL ندارند. این امر نیازمند بازنویسی یا افزودن حجم زیادی از کدهای CSS برای تنظیم چینش عناصر، تراز متن، و جهت‌گیری کلی رابط کاربری است.
- سازگاری با فونت‌های فارسی: انتخاب و افزودن فونت‌های فارسی مناسب که با طراحی کلی قالب همخوانی داشته باشند و در عین حال به درستی در مرورگرهای مختلف نمایش داده شوند، یک چالش دیگر بود.
- مشکلات ظاهری غیرمنتظره: پس از اعمال RTL، ممکن است برخی عناصر به درستی نمایش داده نشوند یا تداخل پیدا کنند که نیازمند دیباگینگ و رفع اشکال دقیق است.

- چالش‌های مربوط به ترجمه و Text Domain:

- عدم رعایت Internationalization: برخی قالب‌ها یا افزونه‌ها به درستی Internationalization (آماده‌سازی برای ترجمه) را رعایت نکرده‌اند. این باعث می‌شود که رشته‌های متنی زیادی در کدها به صورت مستقیم نوشته شده باشند و امکان ترجمه آسان آن‌ها با ابزارهایی مانند Poedit یا Loco Translate وجود نداشته باشد. این امر نیاز به تغییرات مستقیم در کد PHP یا HTML را ایجاد می‌کند که می‌تواند ریسک‌پذیر باشد.
- ترجمه‌ی اصطلاحات تخصصی: ترجمه برخی اصطلاحات فنی یا تخصصی که در قالب‌های فروشگاه‌ی یا گردشگری به کار رفته‌اند، نیاز به دانش تخصصی و یافتن معادل‌های مناسب فارسی داشت.

- چالش‌های مربوط به درگاه‌های پرداخت:

- فقدان ماژول بومی: قالب‌ها و افزونه‌های بین‌المللی طبیعتاً از درگاه‌های پرداخت ایرانی پشتیبانی نمی‌کنند. این امر نیازمند توسعه‌ی یک افزونه‌ی پرداخت اختصاصی یا ادغام یک افزونه‌ی موجود با منطق پرداخت قالب است.
- پیچیدگی API ها: هر درگاه پرداخت ایرانی API ها و مستندات خاص خود را دارد که ادغام آن‌ها نیازمند مطالعه و پیاده‌سازی دقیق است.

- چالش‌های مربوط به لایسنس و تحریم:

- مدیریت لایسنس‌های بین‌المللی: برای محصولاتی که به صورت قانونی از ThemeForest خریداری می‌شوند، مدیریت لایسنس‌های بین‌المللی و استفاده از آن‌ها در محیط ایران به دلیل تحریم‌ها می‌تواند چالش‌برانگیز باشد.
- لایسنس‌گذاری برای مارکت ایرانی: پس از بومی‌سازی، نیاز به اعمال یک سیستم لایسنس‌گذاری مناسب برای عرضه در مارکت پلیس‌های داخلی وجود دارد.

- چالش‌های آماده‌سازی برای مارکت پلیس‌های ایرانی:

- استانداردهای محتوایی: مارکت پلیس‌هایی مانند راست‌چین و ژاکت دارای استانداردهای خاصی برای محتوای متنی و تصویری صفحه محصول هستند که باید با دقت رعایت شوند تا محصول پذیرفته شود.
- تست و تایید: فرآیند تست و تایید محصول توسط تیم مارکت پلیس ممکن است زمان‌بر باشد و نیاز به رفع اشکالات احتمالی بر اساس بازخورد آن‌ها دارد.

غلبه بر این چالش‌ها نیازمند ترکیبی از دانش فنی، تجربه عملی و صبر است. این تحقیق با مستندسازی این چالش‌ها و ارائه راهکارهای عملی، به کاهش این موانع برای توسعه‌دهندگان آتی کمک می‌کند.

۴-۳) پیشنهادهایی برای آینده

با توجه به یافته‌های این تحقیق و چالش‌های مطرح‌شده در فرآیند بومی‌سازی قالب‌ها و افزونه‌های وردپرس برای مارکت پلیس‌های ایرانی، پیشنهاداتی برای بهبود این فرآیند و تحقیقات آتی در این حوزه ارائه می‌شود:

۳-۴-۱) خودکارسازی فرآیند بومی‌سازی

- توسعه ابزارهای خودکارسازی: ایجاد اسکریپت‌ها یا نرم‌افزارهایی که بتوانند بخش‌هایی از فرآیند بومی‌سازی (مانند جایگزینی خودکار کتابخانه‌های تاریخ‌نگار با نسخه‌های جلالی، یا اعمال خودکار تغییرات RTL در CSS) را خودکار کنند. این ابزارها می‌توانند زمان و خطای انسانی را به شدت کاهش دهند.

- استفاده از هوش مصنوعی در ترجمه: بررسی امکان استفاده از مدل‌های پیشرفته‌ی ترجمه ماشینی مبتنی بر هوش مصنوعی که بتوانند ترجمه‌های فرهنگی‌تر و دقیق‌تری برای متون وردپرسی تولید کنند و سپس با ویرایش انسانی تکمیل شوند.
- سیستم‌های یکپارچه بومی‌سازی: توسعه یک پلتفرم یا افزونه‌ی جامع که بتواند تمامی مراحل بومی‌سازی را (از ترجمه تا RTL و جایگزینی تاریخ‌ها) در یک محیط یکپارچه مدیریت کند.

۴-۳-۲) ساخت قالب‌ها و افزونه‌های اختصاصی بومی

- طراحی از ابتدا با در نظر گرفتن نیازهای بومی: به جای بومی‌سازی محصولات بین‌المللی، توسعه‌دهندگان ایرانی می‌توانند به سمت طراحی و توسعه قالب‌ها و افزونه‌هایی حرکت کنند که از ابتدا با در نظر گرفتن نیازهای بازار ایران (تقویم شمسی، RTL بومی، درگاه‌های پرداخت، فونت‌های فارسی و...) ساخته شده‌اند. این رویکرد می‌تواند به محصولات با کیفیت‌تر و با عملکرد بهتر منجر شود.
- توسعه کتابخانه‌ها و ابزارهای بومی استاندارد: جامعه‌ی توسعه‌دهندگان وردپرس فارسی می‌تواند در جهت ایجاد و استانداردسازی کتابخانه‌ها و ابزارهای بومی (مانند یک کتابخانه jdatetime جامع و یکپارچه با وردپرس، یا ابزارهای مدیریت فونت‌های فارسی) تلاش کند تا روند توسعه برای همه ساده‌تر شود.

۴-۳-۳) بهبود همکاری با مارکت پلیس‌ها

- ارائه مستندات جامع‌تر توسط مارکت پلیس‌ها: مارکت پلیس‌های ایرانی می‌توانند مستندات و راهنماهای فنی خود را در زمینه بومی‌سازی و الزامات عرضه محصول، جامع‌تر و به‌روزتر کنند.
- ایجاد بستر تست خودکار: مارکت پلیس‌ها می‌توانند ابزارهای تست خودکار برای بررسی جنبه‌های کلیدی بومی‌سازی (مانند RTL، تقویم شمسی، عملکرد درگاه پرداخت) را در اختیار توسعه‌دهندگان قرار دهند تا فرآیند تایید سریع‌تر و کارآمدتر شود.
- کارگاه‌ها و آموزش‌ها: برگزاری کارگاه‌های آموزشی و وبینارها توسط مارکت پلیس‌ها برای آموزش توسعه‌دهندگان درباره بهترین شیوه‌ها در بومی‌سازی و عرضه محصولات.

۴-۳-۴) تحقیقات آتی در حوزه بومی سازی وردپرس

- مطالعه تطبیقی ابزارهای بومی سازی: انجام یک مطالعه تطبیقی دقیق تر بین ابزارهای مختلف بومی سازی موجود (Poedit, Loco Translate و...) و مقایسه کارایی آنها برای زبان فارسی.
- بررسی عملکرد قالب های بومی سازی شده: تحلیل عملکرد (Performance) قالب های بومی سازی شده در مقایسه با نسخه های اصلی آنها، و شناسایی عوامل مؤثر بر کاهش یا افزایش سرعت.
- مطالعه موردی بر روی افزونه ها: انجام مطالعه موردی مشابهی بر روی بومی سازی یک افزونه پیچیده (مثلاً یک افزونه ووکامرس یا LMS) و بررسی چالش های خاص آن.
- بررسی جنبه های حقوقی و لایسنس گذاری: تحقیق بیشتر در مورد جنبه های حقوقی و چالش های لایسنس گذاری برای محصولات وردپرسی در ایران، به خصوص در مواجهه با تحریم ها.

در نهایت، این پایان نامه نشان داد که بومی سازی قالب ها و افزونه های وردپرس برای بازار ایران، فرآیندی پیچیده اما ضروری است که با برنامه ریزی دقیق و استفاده از ابزارهای مناسب، قابل انجام است. با ادامه تحقیقات و همکاری فعال جامعه توسعه دهندگان و مارکت پلیس ها، می توان به ارتقاء هرچه بیشتر کیفیت و دسترسی پذیری محصولات وردپرسی در کشور کمک شایانی کرد.

فهرست منابع

[۱] قالب ThemeForest (۱۴۰۴). «Traveler – Travel & Tour Booking WordPress»
ThemeForest. از: <https://themeforest.net/item/traveler-traveltourbooking-wordpress-theme/10822683>

[۲] سایت راست چین (۱۴۰۴). «چگونه فروشنده شوید؟». از: <https://www.rtl-theme.com/become-vendor/>

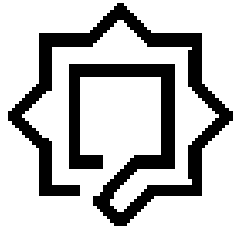
Abstract

In the modern era, websites play a pivotal role in communication, commerce, and everyday life. WordPress, as the most popular content management system (CMS) worldwide, provides a powerful and flexible platform for developing various types of websites. However, many WordPress themes and plugins are developed with a focus on international markets, which creates challenges in terms of compatibility with the needs of Persian-speaking users and the Iranian cultural context. This thesis aims to provide a comprehensive analysis and a practical model for the localization process of WordPress themes and plugins for distribution on Iranian online marketplaces.

The research methodology combines literature review with a practical case study. In the practical section, a popular international theme named *Traveler* was selected, and the complete localization process—from setting up a local development environment (using Laragon) to final preparation for release—was implemented and documented step by step. Key actions included full localization and Jalali (Persian) date adaptation by replacing *daterangepicker* and *moment.js* with their localized versions and integrating the *jdatetime* library into the PHP logic; complete RTL (right-to-left) styling and UI adaptation by modifying CSS and adding Persian fonts; thorough translation of all textual content and post types; and the development of solutions for integrating local payment gateways (such as IDPay) and managing international license challenges.

The findings of this study revealed that localization is far more than mere translation. It requires deep technical knowledge in client-side and server-side programming (JavaScript, PHP, CSS), as well as a strong understanding of WordPress architecture and local user needs. Major challenges, such as the complexity of integrating the Jalali calendar in both frontend and backend, the necessity of extensive CSS modifications for proper RTL support, and the lack of proper internationalization support in some international products were identified and addressed.

This research provides a practical guide and thorough documentation of the challenges involved, supporting Iranian developers in producing high-quality WordPress products that meet the standards of domestic marketplaces such as *Zhakat* and *RTL-Theme*. Suggestions for future work include automating parts of the localization process, developing native themes and plugins from scratch, and improving collaboration with local marketplaces to enrich the Persian WordPress ecosystem and enhance user access to efficient tools.



University of Applied Science and Technology
Iran Informatics Institute of Higher Education

Undergraduate Thesis (B.S.)
Field of Study: Web Programming – Applied Technology
Engineering

TITLE:
A Study on the Localization Process of WordPress Themes and
Plugins for the Iranian Market

Supervisor:
N. Hamrahi Eng.

Author:
Ali Satarzadeh

Spring 2025