

Автор также высказывает предположение, что внедрение технологий ИИ может существенно изменить картину психологической науки, приведя к появлению цифровых двойников когнитивной системы и личностных конструктов[3]. В практическом плане, по мнению автора, могут возникнуть технологии, которые оценивают психологическое состояние человека, помогают его улучшать и даже применять психотерапевтические методы, а также способствуют совершенствованию отношений и ориентируют в поступающей информации. «В фундаментальном плане возможно появление цифровых двойников когнитивной системы, личностных конструктов и т. д. В практическом плане могут возникнуть технологии, которые оценивают психологическое состояние человека и помогают его улучшать вплоть до применения психотерапевтических методов». Можно сделать вывод, ИИ способен к индукции закономерностей, однако эта индукция осуществляется в границах модели, заложенной разработчиком. При этом наличие такой преданной модели далеко не всегда рефлексивируется разработчиком, что и создает впечатление «эмерджентной» познавательной способности нейросети. Ушаков Д.В., в частности, подчеркивает важность осознания контекста и рамок, заданных разработчиком, при использовании ИИ в научных исследованиях.

Научные источники указывают на значительный и многообещающий потенциал технологий искусственного интеллекта для развития психологии. В статье демонстрируется, как ИИ уже сейчас применяется в практической психологии для оказания помощи и поддержки людям, а также поднимаются вопросы о более глубоком влиянии ИИ на научную методологию и понимание когнитивных процессов. Необходимо учитывать возможные риски и этические проблемы, связанные с применением ИИ в психологии, и проводить системный анализ последствий внедрения новых технологий. Для успешной интеграции ИИ в психологическую практику требуется разработка четких этических норм и стандартов, а также обучение специалистов по работе с новыми технологиями. Только в этом случае можно будет реализовать весь потенциал ИИ для улучшения психического здоровья и благополучия людей, а также для углубления нашего понимания человеческого разума и поведения.

Список использованных источников:

1. Всемирная организация здравоохранения. – URL: <https://www.who.int/ru> (дата обращения: 07.01.2023).
2. Корж Е.М. Потенциал применения технологий искусственного интеллекта в психологии / Е.М. Корж, А.В. Громова // Системная психология и социология. – 2023. – № 2 (46). – С. 60–70.
3. Ушаков Д.В. Искусственный интеллект как инструмент психологического исследования / Д.В. Ушаков // Сибирский психологический журнал. – 2023. – № 90. – С. 188–200.

СОВРЕМЕННЫЕ ФРЕЙМВОРКИ И БИБЛИОТЕКИ ДЛЯ ВЕБ-РАЗРАБОТКИ

*П.П. Цветкова², ученица 9 класса,
Научный руководитель: Разумников С.В.^{1а}, к.т.н., доц.
¹Юргинский технологический институт (филиал)*

*Национального исследовательского Томского политехнического университета
652055, Кемеровская обл., г. Юрга, ул. Ленинградская, 26
²МБОУ «СОШ № 14 имени К.С. Федоровского»
652057, Кемеровская обл., г. Юрга, ул. Волгоградская, 3
E-mail: razumnikov@tpu.ru*

Аннотация: В статье рассматриваются современные фреймворки и библиотеки для веб-разработки.

Ключевые слова: веб-разработка, библиотека, фреймворк.

Abstract: The article discusses modern frameworks and libraries for web development.

Keywords: web development, library, framework.

Введение. Веб-разработка постоянно эволюционирует, и новые тренды появляются практически ежегодно. Рассмотрим некоторые из ключевых направлений, которые будут актуальны в ближайшие годы:

1. Jamstack

Jamstack – это архитектурный подход к созданию веб-сайтов и приложений, основанный на использовании предварительно сгенерированных страниц (JAM = JavaScript, APIs, Markup). Этот подход предполагает, что все статические части сайта собираются заранее, а взаимодействие с пользователем обеспечивается через JavaScript и API.

Jamstack позволяет значительно улучшить производительность, безопасность и масштабируемость веб-проектов. Статически сгенерированные сайты быстрее загружаются, легче кэшируются и требуют меньше ресурсов сервера. Примеры инструментов: Next.js, Gatsby, Nuxt.js, Hugo.

2. Serverless Computing

Serverless (безсерверные вычисления) – это модель облачных вычислений, в которой поставщик облачных услуг управляет выделением машинных ресурсов, а разработчики фокусируются исключительно на написании кода и бизнес-логике. Serverless позволяет снизить затраты на инфраструктуру и обслуживание, а также ускорить разработку и развертывание приложений. Вам не нужно беспокоиться о настройке серверов, их масштабировании и мониторинге. Примеры сервисов: AWS Lambda, Azure Functions, Google Cloud Functions.

3. Progressive Web Apps (PWA)

PWA – это веб-приложения, которые сочетают в себе функциональность нативных мобильных приложений и удобство доступа через браузер.

Они могут работать офлайн, отправлять push-уведомления и устанавливаться на домашний экран устройства. PWA обеспечивают лучшее пользовательское восприятие, увеличивают вовлеченность пользователей и снижают барьеры для установки приложений. Кроме того, они работают на любых устройствах и операционных системах. Примеры инструментов: Workbox, Lighthouse, Service Workers.

4. Headless CMS

Headless CMS – это контент-менеджмент система, которая отделяет фронтэнд от бэкэнда. Контент хранится и управляется отдельно, а доступ к нему осуществляется через API. Headless CMS дает возможность использовать один источник контента для нескольких каналов доставки (веб-сайт, мобильное приложение, IoT-устройства и др.). Это увеличивает гибкость и ускоряет внедрение изменений. Примеры систем: Strapi, Contentful, GraphCMS.

5. Micro Frontends

Micro Frontends – это подход к разработке фронтенда, при котором приложение делится на небольшие, независимые части, каждая из которых может разрабатываться и развертываться отдельной командой. Micro Frontends помогают упростить управление большими проектами, уменьшить зависимость между командами и ускорить выпуск новых функций. Это особенно полезно для крупных компаний с распределёнными командами. Module Federation (Webpack), Single-spa.

6. Artificial Intelligence (AI) и Machine Learning (ML)

AI и ML применяются для улучшения взаимодействия с пользователями, персонализации контента, автоматизации процессов и анализа данных. Интеграция AI и ML в веб-проекты позволяет предоставлять пользователям более релевантный опыт, улучшать конверсию и повышать лояльность клиентов. Например, чат-боты на основе AI могут отвечать на вопросы пользователей круглосуточно, а рекомендательные системы на основе ML помогут предложить товары или услуги, соответствующие интересам клиента. Примеры применения: ChatGPT, TensorFlow.js, Amazon Personalize.

7. WebAssembly (Wasm)

Wasm – это низкоуровневый бинарный формат, который позволяет выполнять код, написанный на языках программирования C/C++, Rust, Go и других, прямо в браузере с почти родной скоростью. Wasm открывает новые горизонты для веб-разработки, позволяя переносить сложные вычислительные задачи в браузер. Это может существенно повысить производительность веб-приложений и расширить их функциональные возможности. Примеры использования: Blazor, Emscripten, AssemblyScript.

8. Composable Architecture

Composable architecture – это подход к разработке программного обеспечения, при котором приложение строится из независимых, взаимозаменяемых частей (компонентов). Эти компоненты могут быть собраны вместе для достижения нужной функциональности. Composable architecture помогает обеспечить гибкость и устойчивость к изменениям. Разделение на компоненты упрощает тестирование, обновление и замену отдельных частей приложения без влияния на другие. The Composable Architecture (Swift), Redux Toolkit.

9. Accessibility (Доступность)

Доступность означает создание веб-сайтов и приложений, которые могут быть использованы всеми людьми, независимо от их физических ограничений или технических возможностей. Повышение доступности не только улучшает пользовательский опыт для людей с ограниченными возможностями, но и соответствует законодательным требованиям многих стран. Это также способствует улучшению SEO и увеличению аудитории.

10. Dark Mode

Dark mode – это режим отображения интерфейса, при котором фон становится тёмным, а текст светлым. Это снижает нагрузку на глаза при работе в условиях низкой освещенности и экономит заряд батареи на некоторых устройствах. Пользователи всё чаще предпочитают dark mode, поскольку он уменьшает усталость глаз и выглядит современно. Многие крупные компании уже реализовали эту функцию в своих продуктах.

Для фронтенд-разработки существует несколько современных библиотек, каждая из которых обладает своими уникальными особенностями и преимуществами. Вот некоторые из них:

1. React.js. Одна из самых популярных библиотек для создания пользовательских интерфейсов. React.js позволяет разбивать интерфейс на компоненты, что делает код более управляемым и переиспользуемым. Особенности:

- Виртуальный DOM для высокой производительности.
- Компоненты позволяют создавать модули, которые можно многократно использовать.
- Поддерживает JSX – синтаксический сахар для написания HTML внутри JavaScript.
- Широкая экосистема и активные сообщества.

2. Vue.js Прогрессивный фреймворк для создания пользовательских интерфейсов. Vue.js известен своей легкостью и гибкостью, что делает его отличным выбором для быстрого старта новых проектов. Особенности:

- Простое начало работы даже для начинающих разработчиков.
- Легкая интеграция в уже существующие проекты.
- Реактивная система данных.
- Хорошая производительность и малый вес.

3. Svelte – это компилируемый фреймворк, который отличается тем, что он генерирует чистый JavaScript без необходимости в runtime-библиотеке. Особенности:

- Минимальное влияние на производительность, так как большая часть логики выполняется на этапе сборки.

- Меньший размер конечного файла по сравнению с React и Vue.
- Интуитивно понятный синтаксис и простота в изучении.

Выбор библиотеки зависит от ваших потребностей и опыта. Для большинства проектов React и Vue остаются самыми популярными вариантами благодаря их широкой поддержке и сообществу. Однако, если вам нужна максимальная производительность или легкость интеграции, стоит рассмотреть Svelte, Preact или Alpine.js.

Список использованных источников:

1. Ломайкин А.С. Роль фреймворков и библиотек в современной веб-разработке / А.С. Ломайкин // Материалы XXIV научно-практической конференции молодых ученых, аспирантов и студентов национального исследовательского мордовского государственного университета им. Н.П. Огарёва. – 2021. – С. 268–273.

2. Омельченко Д.А. Современные тенденции веб-разработки: front-end фреймворки / Д.А. Омельченко, Р.Г. Гонатаев, Е.А. Иванова // TECHNICAL SCIENCE. – 2020. – 48 с.

ВЛИЯНИЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА НА РЫНОК ТРУДА В ПРОМЫШЛЕННОСТИ

Т.С. Советбеков^а, С.П. Богаченко, студенты гр. 17В41,

Научный руководитель: Телипенко Е.В., к.т.н., доц.

Юргинский технологический институт (филиал)

Национального исследовательского Томского политехнического университета

652055, Кемеровская обл., г. Юрга, ул. Ленинградская, 26

E-mail: ^аsovetbekovtemirlan4@gmail.com

Аннотация: В данной статье рассматривается влияние искусственного интеллекта (ИИ) на рынок труда в промышленности. Анализируются положительные и отрицательные аспекты внедрения ИИ, изменения в структуре занятости и перспективы адаптации работников к новым условиям. Рассматриваются примеры успешного внедрения ИИ в промышленности, а также методы переквалификации сотрудников для работы в новой технологической среде.