

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

**Домашняя работа 5: изучение основ работы с менеджером
зависимостей прм**

Выполнил:
Фирсов Илья
Группа К3441
Проверил:
Добряков Д. И.

Санкт-Петербург
2026 г.

Цель

В рамках данной работы Вам предстоит изучить основные команды пакетного менеджера NPM и научиться стартовать проект на Vue.

Научиться работать с npm и vue на основе мануала: <https://docs.google.com/document/d/187UkgGNrcWqkb2aCGpkHTLgeozoElMqdVgVGMB0C9gk/edit?usp=sharing>.

Ход работы

1. Инициализация проекта - выполнил `npm create vite@latest vacancy-manual -- --template vue`, затем обновил `package.json`, чтобы добавить `start` на 8080, `dev`, `build`, `preview` и `lint` (eslint).
2. Установка зависимостей - `npm install axios pinia pinia-persists bootstrap` и `npm install --save-dev eslint`. npm создал `package-lock.json`.
3. Настройка глобальных пакетов - `src/main.js` теперь импортирует `router`, `pinia`, Bootstrap-стили и `assets/main.css`. `src/stores/index.js` создаёт Pinia и подключает `pinia-persists`. `vite.config.js` добавил алиас `@`, чтобы импорты были короткими (`@/api`, `@/views` и т.п.).
4. API/стор - добавлен `src/api/instance.js` (axios-инстанс с базовым URL `http://localhost:3000`), `VacanciesApi` (`src/api/vacancies.js`) и `vacanciesApi` в `src/api/index.js`. `src/stores/vacancies.js`: загрузка `vacancies`, обработка ошибок с fallback-данными, создание вакансии через API или локально, всё через Pinia-прерванный `persist`.
5. Вёрстка страницы вакансий - `VacanciesPage.vue` может добавлять вакансии, перезагружать список и отображать карточки `VacancyCard` с названием, компанией, локацией, зарплатой (форматирования и метки). Этот компонент обёрнут в `BaseLayout.vue`, а `VacancyCard` (файл `src/components/VacancyCard.vue`) выводит описание и дату публикации.
6. Сборка/проверка - запуск `npm run build` внутри `vacancy-manual` показал корректную сборку (модули, `dist`). Команда `npm start`/`npm run dev` готова поднимать интерфейс для ручной проверки.

Команды

- `npm create vite@latest vacancy-manual -- --template vue`
- `npm install`
- `npm install axios pinia pinia-persists bootstrap`
- `npm install --save-dev eslint`
- `npm run build` (внутри `vacancy-manual`)
- `npm start` / `npm run dev` (для ручного теста)

- `src/views/VacanciesPage.vue`: форма `@submit.prevent="createCard"`, `v-model`, кнопка "Сохранить вакансию", список `VacancyCard`.

```
<form class="row g-3" @submit.prevent="createCard">
  <input v-model="form.title" type="text" class="form-control" required />
  <input v-model="form.company" type="text" class="form-control" required />
  ...
</form>
<div class="row g-3">
  <div class="col-md-6" v-for="vacancy in store.vacancies" :key="vacancy.id">
    <VacancyCard :vacancy="vacancy" />
  </div>
</div>
```

- `src/stores/vacancies.js`: `loadVacancies()` загружает данные через `vacanciesApi`, `createVacancy()` добавляет запись и вставляет её в начало списка.

- `src/components/VacancyCard.vue`: отображает заголовок, компанию, локацию, зарплату и дату публикации; `salaryLabel` форматируется через `computed`.

- `src/api/instance.js` + `src/api/vacancies.js` реализуют Axios-инстанс и класс API-помощника, как в мануале.

- `src/stores/index.js`: подключение Pinia и `pinia-persists`, чтобы `vacancies` хранился в `localStorage`.

Примеры кода

1. `vacancy-manual/src/views/VacanciesPage.vue:1-100` - форма, коллекция карточек и реактивный стор, всё в одном представлении:

```
<form class="row g-3" @submit.prevent="createCard">
  <input v-model="form.title" type="text" class="form-control" required />
  <input v-model="form.company" type="text" class="form-control" required />
  ...
</form>
<div class="row g-3">
  <div class="col-md-6" v-for="vacancy in store.vacancies" :key="vacancy.id">
    <VacancyCard :vacancy="vacancy" />
  </div>
</div>
```

2. `vacancy-manual/src/stores/vacancies.js:1-60` - Pinia store с API-загрузкой и fallback-данными плюс создание новых записей:

```
export const useVacanciesStore = defineStore('vacancies', {
  state: () => ({ vacancies: [] }),
  actions: {
    async loadVacancies() {
      const response = await vacanciesApi.getAll()
      this.vacancies = Array.isArray(response?.data) ? response.data : fallbackVacancies
    },
    async createVacancy(vacancy) {
```

```

const payload = { ...vacancy, id: Date.now().toString(), postedAt: 'только что' }
this.vacancies.unshift(payload)
},
}
})

```

3. `vacancy-manual/src/components/VacancyCard.vue:1-30` - обёртка карточки вакансии со стилизацией и computed-зарплатой:

```

```vue
<article class="card vacancy-card h-100">
 <div class="card-body">
 <h3 class="h5 mb-2">{{ vacancy.title }}</h3>
 <p class="text-muted mb-1">{{ vacancy.company }}</p>
 <p class="badge bg-primary text-light mb-2">{{ vacancy.location || 'Remote' }}</p>
 {{ salaryLabel }}
 </div>
</article>

```

Заполненная форма и новое объявление в списке:

**Career Atlas · страница вакансий**

Форма создания вакансии

**Добавить вакансию**

Название	Компания
дизайнер	some
Локация	Зарплата (руб.)
Сочи	250000
Описание	
описание	

**Сохранить вакансию**

**Обновить**

<b>frontend разработчик</b> some <b>Спб</b> описание <b>3 000 000 ₽</b>	<b>backend разработчик</b> some <b>Москва</b> описание <b>300 000 ₽</b>
-------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------

только что

# Career Atlas · страница вакансий

Форма создания вакансии

## Добавить вакансию

Название

Компания

Локация

Зарплата (руб.)

Описание

[Сохранить вакансию](#)

[Обновить](#)

дизайнер

some

[Сочи](#)

описание

250 000 ₽

frontend разработчик

some

[Спб](#)

описание

3 000 000 ₽

только что

## Выход

- npm позволяет получить единое окружение: `npm install` фиксирует зависимости, `npm run dev`/`start` поднимают Vite, `npm run build` создаёт `dist`, `npm run lint` проверяет стандарты.
- Проект готов к запуску (запускается `npm start`, axios сохраняет/загружает вакансии, bootstrap отвечает за стили).