

```

<template>
  <v-row justify="center" align="center">
    <v-col cols="12" sm="8" md="6">
      <v-text-field
        v-model="order_number"
        readonly
        label="Номер заказа"
      />
      <v-text-field
        v-model="client"
        label="Заказчик"
      />
      <v-text-field
        v-model="phone"
        v-mask="' +375 ## ### ## #' "
        label="Телефон заказчика"
      />
      <v-text-field
        v-model="adress"
        label="Адрес"
      />
      <v-menu
        ref="menu"
        v-model="dateShow"
        :close-on-content-click="false"
        transition="scale-transition"
        offset-y
        min-width="290px"
      >
        <template v-slot:activator="{ on, attrs }">
          <v-text-field
            v-model="computedDateFormatted"
            label="Дата готовности"
            prepend-icon="mdi-calendar"
            readonly
            v-bind="attrs"
            v-on="on"
          />
        </template>
        <v-date-picker
          ref="picker"
          v-model="date_of_ready"
          @change="save"
        />
      </v-menu>
    </v-col>
    <v-col cols="9">
      <v-select
        v-model="discount_name"
        :items="discountsList"
        label="Скидка"
        item-text="discount_name"
      />
    </v-col>
  </v-row>

```

```

<v-col cols="3">
  <v-text-field
    v-model="discount.discount_percent"
    readonly
    label="Скидка, %"
  />
</v-col>
</v-row>
<v-row>
  <v-col cols="3">
    <v-btn
      style="margin-top: 15px"
      @click="minus"
    >
      убрать
    </v-btn>
  </v-col>
  <v-col cols="6">
    <v-text-field
      v-model="products.length"
      label="Изделие №"
    />
  </v-col>
  <v-col cols="3">
    <v-btn
      style="margin-top: 15px"
      @click="plus"
    >
      добавить
    </v-btn>
  </v-col>
</v-row>
<v-card v-for="(pr, index) in products" :key="'product' + index"
style="margin-bottom: 24px">
  <v-card-text>
    <v-select
      v-model="pr.product_type"
      :items="dataOrder.products"
      label="Тип изделия"
      item-text="products_types"
    />
  <v-row>
    <v-col cols="4">
      <v-row>
        <v-col cols="2">
          <div style="width: 20px; height: 20px; margin-top: 20px"
:style="{background: pr.color}" />
        </v-col>
        <v-col cols="10">
          <v-select
            v-model="pr.color"
            :items="dataOrder.colors"
            label="Цвет"
            item-text="color_name"
          >

```

```

        item-value="color"
      />
    </v-col>
  </v-row>
</v-col>
<v-col cols="4" style="margin-top: 12px;">
  <v-text-field
    v-model="pr.length"
    label="Длина"
  />
</v-col>
<v-col cols="4" style="margin-top: 12px;">
  <v-text-field
    v-model="pr.width"
    label="Ширина"
  />
</v-col>
</v-row>
<v-combobox
  v-model="pr.defects"
  :items="dataOrder.defects"
  label="Дефекты"
  small-chips
  multiple
  dense
/>
<v-row>
  <v-col cols="9" style="margin-top: 12px;">
    <v-select
      v-model="pr.service_name"
      :items="services[pr.product_type]"
      label="Услуга"
      item-text="service_name"
    />
  </v-col>
  <v-col v-if="curServices[pr.product_type][pr.service_name]" cols="3"
style="margin-top: 12px;">
    <v-text-field
      :value="curServices[pr.product_type][pr.service_name].service_price + 'р.'"
      readonly
      label="Цена"
    />
  </v-col>
</v-row>
<v-row>
  <v-col cols="3">
    <v-currency-field
      v-model="pr.product_price"
      label="Стоимость"
    />
  </v-col>
  <v-col cols="3">
    <v-text-field

```

```

        v-model="pr.product_wearout"
        label="Износ, %"
    />
</v-col>
<v-col cols="6">
    <v-select
        v-model="pr.products_poll"
        :items="dataOrder.product_polls"
        label="Загрязнение"
    />
</v-col>
</v-row>
<v-text-field
    v-model="pr.comment"
    label="Примечание"
/>
</v-card-text>
</v-card>
<v-row>
    <v-col cols="5">
        <v-currency-field
            :value="SUM"
            suffix="р."
            readonly
            label="Сумма"
        />
    </v-col>
    <v-col cols="2">
        <v-icon style="position: relative; top: 24px; left: 24px;" @click="PAID
= SUM">
            mdi-arrow-right-bold
        </v-icon>
    </v-col>
    <v-col cols="5">
        <v-currency-field
            v-model="PAID"
            suffix="р."
            label="Оплачено"
        />
    </v-col>
</v-row>
</v-row>
<v-row>
    <v-col cols="4" justify="center" align="center">
        <v-btn
            color="accent"
            @click="saveOrder"
        >
            сохранить
        </v-btn>
    </v-col>
    <v-col cols="4" justify="center" align="center">
        <v-btn
            @click="printOrd"
        >

```

```

        печать
    </v-btn>
</v-col>
<v-col cols="4" justify="center" align="center">
    <v-btn
        color="red"
        @click="deleteOrd"
    >
        удалить
    </v-btn>
</v-col>
</v-row>
</v-col>
<v-snackbar
    v-model="exitSnack"
>
    Выйти без сохранения?

    <template v-slot:action="{ attrs }">
        <v-btn
            text
            v-bind="attrs"
            @click="exitSnack = false"
        >
            Отмена
        </v-btn>
        <v-btn
            color="red"
            text
            v-bind="attrs"
            @click="$router.push('/actions')"
        >
            Выйти
        </v-btn>
    </template>
</v-snackbar>
<v-snackbar
    v-model="message_snack_status"
>
    {{ snack_message }}
    <template v-slot:action="{ attrs }">
        <v-btn
            text
            v-bind="attrs"
            @click="message_snack_status = false"
        >
            Закрыть
        </v-btn>
    </template>
</v-snackbar>
</v-row>
</template>
<script>
export default {

```

```

name: 'NewOrder',
data () {
  return {
    dateShow: false,
    exitSnack: false,
    message_snack_status: false,
    snack_message: '',

    client: undefined,
    phone: undefined,
    adress: undefined,
    date_of_ready: undefined,
    discount_name: undefined,
    products: [
    ],
    PAID: undefined
  }
},
computed: {
  computedDateFormatted () {
    return this.formatDate(this.date_of_ready)
  },

  dataOrder () {
    return this.$store.state.data
  },
  order_number () {
    return this.$store.state.order_number
  },
  discountsList () {
    return [
      {
        discount_name: 'без скидки',
        discount_percent: '0'
      }
    ].concat(this.dataOrder.discounts)
  },
  discount () {
    if (this.discountsList && this.discountsList.filter(item => item &&
item.discount_name === this.discount_name).length) {
      return this.discountsList.filter(item => item.discount_name ===
this.discount_name)[0]
    } else {
      return {
        discount_name: undefined,
        discount_percent: undefined
      }
    }
  },
  SUM () {
    return this.products.reduce((sum, item) => {
      const service = this.services[item.product_type].filter(ser =>
ser.service_name === item.service_name)[0]
      if (service) {

```

```

        const price = service.service_price
        switch (service.price_method) {
            case 'square':
                return (parseFloat(sum) + (item.length.replace(/,//, '.') *
item.width.replace(/,//, '.') * price * (1 - this.discount.discount_percent /
100))).toFixed(2)
            case 'perimeter':
                return (parseFloat(sum) + ((item.length.replace(/,//, '.') +
item.width.replace(/,//, '.')) * price * (1 - this.discount.discount_percent /
100))).toFixed(2)
            case 'line':
                return (parseFloat(sum) + (item.length.replace(/,//, '.') * price *
(1 - this.discount.discount_percent / 100))).toFixed(2)
            default:
                return sum + 0
        }
    }
    return sum + 0
}, 0)
},
services () {
    return this.dataOrder.products.reduce((arr, item) => {
        arr[item.products_types] = item.services
        return arr
    }, [])
},
curServices () {
    return this.dataOrder.products.reduce((arr, item) => {
        arr[item.products_types] = item.services.reduce((arr2, item2) => {
            arr2[item2.service_name] = item2
            return arr2
        }, [])
        return arr
    }, [])
},
printLines () {
    const lines = []
    lines.concat(this.dataOrder.prefix_text.split('\n'))
    lines.push(`Заказ № ${this.order_number} Дата ${this.formatDate((new
Date()).toISOString().substr(0, 10))}`)
    lines.push(`Заказчик ${this.client}`)
    lines.push(`телефон ${this.phone}. К ${this.computedDateFormatted}`)

    this.products.forEach((item) => {
        lines.push(`${item.product_type}`)

        let price = 0
        switch
(this.curServices[item.product_type][item.service_name].price_method) {
            case 'square':
                price = ((item.length.replace(/,//, '.') * item.width.replace(/,//,
'.') * price * (1 - this.discount.discount_percent / 100))).toFixed(2)
                break
            case 'perimeter':

```

```

        price = (((2 * item.length.replace(/,/ , '.') + 2 *
item.width.replace(/,/ , '.')) * price * (1 - this.discount.discount_percent /
100))).toFixed(2)
        break
        case 'line':
            price = ((item.length.replace(/,/ , '.') * price * (1 -
this.discount.discount_percent / 100))).toFixed(2)
            break
        }
        let textColor = this.dataOrder.colors.filter(item => item.color ===
this.color)
        textColor = textColor.length > 0 ? textColor[0].color_name :
'неизвестно'
        lines.push(`${price} X ${item.length} X ${item.width}, ${textColor}`)

        lines.push(`${item.service_name}`)
        lines.push(`${item.product_price}, ${item.product_wearout},
${item.products_poll}`)
        lines.push(`${item.defects.join(', ')}`)
        lines.push(`${item.comment}`)
    })

    lines.push(`Сумма ${this.SUM} Оплачено ${this.PAID}`)
    lines.concat(this.dataOrder.postfix_text.split('\n'))

    return lines
}
},
watch: {
    dateMenu (val) {
        val && setTimeout(() => (this.$refs.picker.activePicker = 'YEAR'))
    }
},
mounted () {
    const unregisterRouterGuard = this.$router.beforeEach((to, from, next) => {
        if (this.exitSnack || this.message_snack_status) {
            next()
        } else {
            this.exitSnack = true
            next(false)
        }
    })
    this.$once('hook:destroyed', () => {
        unregisterRouterGuard()
    })
    if (!this.$store.state.auth) {
        this.$router.push('/service_login')
    }
    const date = (new Date())
    date.setDate(date.getDate() + this.dataOrder.order_duration)
    this.date_of_ready = date.toISOString().substr(0, 10)
    this.discount_name = this.discountsList[0].discount_name

    this.plus()

```

```

},
methods: {
  formatDate (date) {
    if (!date) { return null }

    const [year, month, day] = date.split('-')
    return `${day}.${month}.${year}`
  },
  parseDate (date) {
    if (!date) { return null }

    const [day, month, year] = date.split('.')
    return `${year}-${month.padStart(2, '0')}-${day.padStart(2, '0')}`
  },
  save (date) {
    this.$refs.menu.save(date)
  },
  minus () {
    if (this.products.length > 1) {
      this.products.pop()
    }
  },
  plus () {
    this.products.push({
      product_number: 1,
      product_type: this.dataOrder.products.length ?
this.dataOrder.products[0].products_types : undefined,
      color: '#FFFFFF',
      length: '2.5',
      width: '1',
      service_name: this.dataOrder.products[0].services[0].service_name,
      product_price: this.dataOrder.product_price,
      product_wearout: this.dataOrder.product_wearout,
      products_poll: this.dataOrder.product_polls.length ?
this.dataOrder.product_polls[0] : undefined,
      comment: '',
      defects: []
    })
  },
  saveOrder () {
    this.$axios.post('https://api.kovrochistka.by/api/add_order', {
      login: this.$store.state.data.login,
      password: this.$store.state.data.password,
      order_number: this.$store.state.order_number,
      client: this.client,
      phone: this.phone,
      adress: this.adress,
      date_of_ready: this.computedDateFormatted,
      discount_name: this.discount_name,
      products: this.products,
      SUM: this.SUM,
      PAID: this.PAID
    }).then(() => {
      this.message_snack_status = true
    })
  }
}

```

```
        this.snack_message = 'Сохраненно успешно.'
        this.$router.push('/actions')
    }).catch(() => {
        this.message_snack_status = true
        this.snack_message = 'Ошибка сохранения.'
    })
},
deleteOrd () {
    this.exitSnack = true
},
printOrd () {
    this.$axios.get(this.dataOrder.URL_KSA, {
        headers: {
            'Content-Type': 'application/json'
        },
        params: {
            printBlocks: [{
                kind: 'text',
                lines: this.printLines
            }]
        }
    }).then(() => {
        this.message_snack_status = true
        this.snack_message = 'Отправленно на печать.'
    }).catch(() => {
        this.message_snack_status = true
        this.snack_message = 'Ошибка печати.'
    })
}
}
}
</script>

<style scoped>

</style>
```