```
import Vue from 'vue'
import Vuex from 'vuex'
import axios from './axios-auth'
import globalAxios from 'axios'

import router from './router'

Vue.use(Vuex)

export default new Vuex.Store({
  state: {
    idToken: null,
    userId: null,
    user: null
  },
  mutations: {
    authUser (state, userData) {
      state.idToken = userData.token
      state.userId = userData.userId
    },
    storeUser (state, user) {
      state.user = user
    },
    clearAuthData (state) {
      state.idToken = null
      state.userId = null
    }
  },
  actions: {
    setLogoutTimer ({commit}, expirationTime) {
      setTimeout(() => {
        commit('clearAuthData')
      }, expirationTime * 1000)
    },
    signup ({commit, dispatch}, authData) {
      axios.post('/signupNewUser?key=AIzaSyDQRGLDgM468zVI0ukobObhLo1nH7YKIoQ', {
        email: authData.email,
        password: authData.password,
        returnSecureToken: true
      })
        .then(res => {
          console.log(res)
          commit('authUser', {
            token: res.data.idToken,
            userId: res.data.localId
          })
          const now = new Date()
          const expirationDate = new Date(now.getTime() + res.data.expiresIn * 1000)
          localStorage.setItem('token', res.data.idToken)
          localStorage.setItem('userId', res.data.localId)
          localStorage.setItem('expirationDate', expirationDate)
          router.push('/dashboard')
          dispatch('storeUser', authData)
          dispatch('setLogoutTimer', res.data.expiresIn)
        })
        .catch(error => console.log(error))
    },
    login ({commit, dispatch}, authData) {
      axios.post('/verifyPassword?key=AIzaSyDQRGLDgM468zVI0ukobObhLo1nH7YKIoQ', {
        email: authData.email,
        password: authData.password,
        returnSecureToken: true
      })
        .then(res => {
          console.log(res)
          const now = new Date()
          const expirationDate = new Date(now.getTime() + res.data.expiresIn * 1000)
          localStorage.setItem('token', res.data.idToken)
          localStorage.setItem('userId', res.data.localId)
          localStorage.setItem('expirationDate', expirationDate)
          commit('authUser', {
            token: res.data.idToken,
            userId: res.data.localId
          })
          router.push('/dashboard')
          dispatch('setLogoutTimer', res.data.expiresIn)
        })
        .catch(error => console.log(error))
    },
    tryAutoLogin ({commit}) {
      const token = localStorage.getItem('token')
      if (!token) {
        return
      }
      const expirationDate = localStorage.getItem('expirationDate')
      const now = new Date()
      if (now >= expirationDate) {
        return
      }
      const userId = localStorage.getItem('userId')
      commit('authUser', {
        token: token,
        userId: userId
      })
    },
    logout ({commit}) {
      commit('clearAuthData')
      localStorage.removeItem('expirationDate')
      localStorage.removeItem('token')
      localStorage.removeItem('userId')
      router.replace('/signin')
    },
    storeUser ({commit, state}, userData) {
      if (!state.idToken) {
        return
      }
      globalAxios.post('/users.json' + '?auth=' + state.idToken, userData)
        .then(res => {
          console.log(res.data)
          commit('storeUser', {
            user: res.data
          })
        })
        .catch(error => console.log(error))
    },
    fetchUser ({commit, state}) {
      if (!state.idToken) {
        return
      }
      globalAxios.get('/users.json' + '?auth=' + state.idToken)
        .then(res => {
          console.log(res)
          const data = res.data
          const users = []
          for (let key in data) {
            const user = data[key]
            user.id = key
            users.push(user)
          }
          console.log(users)
          commit('storeUser', users[0])
        })
        .catch(error => console.log(error))
    }
  },
  getters: {
    user (state) {
      return state.user
    },
    isAuthenticated (state) {
      return state.idToken !== null
    }
  }
})
```